Computational Geometry

Convex Hulls in 3D
Lecture #9

[Comp. Geom A&A : Chapter 11]
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$,
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#\text{edges on } \partial \text{CH}(S)$?
# Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. \#edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. #edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>( w\text{-c complexity of } \text{CH}(S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(d)</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>$\text{w-c complexity of CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
</tbody>
</table>

What is max. number of edges on $\partial \text{CH}(S)$?
# Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set \( S \) of \( n \) points in \( \mathbb{R}^d \), what is max. \#edges on \( \partial \text{CH}(S) \)?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 2 \in \Theta(1) )</td>
</tr>
<tr>
<td>2</td>
<td>( n \in \Theta(n) )</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( d )</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>Your task!</td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. #edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial\text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of $\text{CH}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^{\lfloor d/2 \rfloor})$</td>
</tr>
</tbody>
</table>

Upper Bound Theorem
### Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$edges on $\partial \text{CH}(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>w-c complexity of CH($S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^\lfloor d/2 \rfloor)$</td>
</tr>
</tbody>
</table>

**Upper Bound Theorem**

**Construction?**
Complexity of the Convex Hull

Given set $S$ of $n$ points in $\mathbb{R}^d$, what is max. $\#$ edges on $\partial CH(S)$?

<table>
<thead>
<tr>
<th>dim</th>
<th>$w$-c complexity of $CH(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2 \in \Theta(1)$</td>
</tr>
<tr>
<td>2</td>
<td>$n \in \Theta(n)$</td>
</tr>
<tr>
<td>3</td>
<td>$3n - 6 \in \Theta(n)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\Theta(n^\lfloor d/2 \rfloor)$</td>
</tr>
</tbody>
</table>

Upper Bound Theorem

Construction

randomized-incremental!
Visibility
Visibility

Face $f$ is visible from $p$ but not from $q$. 
Face $f$ is *visible* from $p$ but not from $q$. 
Face $f$ is visible from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$. 
Visibility

Face $f$ is visible from $p$ but not from $q$.

Define conflict graph $G$:

- **points (visibility)**
- **facets**

conflicts

- $r$
- $f$
Face $f$ is visible from $p$ but not from $q$.

Define conflict graph $G$:
Visibility

Face $f$ is visible from $p$ but not from $q$.

Define conflict graph $G$:

- **conflicts (visibility)**
  - points
  - facets
  - $P_{\text{conflict}}(f)$
  - $F_{\text{conflict}}(r)$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. ($p_5, \ldots, p_n$) of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$
do

if $\text{conflict} (p_r) \neq \emptyset$ then

delete all facets in $\text{conflict} (p_r)$ from $C$

$L \leftarrow \text{list of horizon edges visible from } p_r$

foreach $e \in L$ do

$f \leftarrow \text{C.create facet}(e, p_r)$; create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow \text{previously incident } C(e)$

$P(e) \leftarrow P(\text{conflict} (f_1)) \cup P(\text{conflict} (f_2))$

foreach $p \in P(e)$ do

if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

delete vtc $\{p_r\} \cup \text{conflict} (p_r)$ from $G$
do

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)
Rand3dConvexHull($P \subseteq \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$:

$(p, f)$ edge $\iff$ $f$ visible from $p$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$:

$(p, f)$ edge $\Leftrightarrow$ $f$ visible from $p$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

\[\text{for } r = 5 \text{ to } n \text{ do}\]

\[\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \}\]

\[\text{return } C\]
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\[ \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \notin C \} \]

return $C$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)
compute rand. perm. \((p_5, \ldots, p_n)\) of \( P \setminus P' \)
initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do

\[ \text{if } \text{F}_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \} \]
delete all facets in \( \text{F}_{\text{conflict}}(p_r) \) from \( C \)

return \( C \)
Rand3dConvexHull($P \subseteq \mathbb{R}^3$)

pick non-coplanar set $P' = \{ p_1, \ldots, p_4 \} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\hspace{1cm} if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

\hspace{1cm} delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \notin C$ \}

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow \text{list of horizon edges visible from } p_r$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. ($p_5, \ldots, p_n$) of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \}
\]

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow \text{list of horizon edges visible from } p_r$

return $C$
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)
\( C \leftarrow \text{CH}(P') \)
compute rand. perm. \((p_5, \ldots, p_n)\) of \( P \setminus P' \)
initialize conflict graph \( G \)

\[
\text{for } r = 5 \text{ to } n \text{ do}
\]
\[
\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \notin C \}
\]
delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)
\( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)

\[
\text{foreach } e \in \mathcal{L} \text{ do}
\]

return \( C \)
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)
pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)
\( C \leftarrow \text{CH}(P') \)
compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)
initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do
  if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then \( \{ p_r \notin C \} \)
    delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)
    \( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)
  foreach \( e \in \mathcal{L} \) do
    \( f \leftarrow C.\text{create\_facet}(e, p_r); \text{create vtx for } f \text{ in } G \)

return \( C \)
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)
pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)
\(C \leftarrow \text{CH}(P')\)
compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)
initialize conflict graph \(G\)
for \(r = 5\) to \(n\) do
  if \(F_{\text{conflict}}(p_r) \neq \emptyset\) then \(\{p_r \not\in C\}\)
  delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)
  \(\mathcal{L} \leftarrow\) list of horizon edges visible from \(p_r\)
  foreach \(e \in \mathcal{L}\) do
    \(f \leftarrow C.\text{create\_facet}(e, p_r)\); create vtx for \(f\) in \(G\)
    \((f_1, f_2) \leftarrow \text{previously\_incident}_C(e)\)
    \(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)
return \(C\)
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \((p_5, \ldots, p_n)\) of \( P \setminus P' \)

initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do

  if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then \( \{ p_r \not\in C \} \)

  delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)

  \( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)

  foreach \( e \in \mathcal{L} \) do

    \( f \leftarrow C.\text{create\_facet}(e, p_r); \text{create vtx for } f \text{ in } G \)

    \( (f_1, f_2) \leftarrow \text{previously\_incident}_C(e) \)

    \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)

return \( C \)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

1. Pick a non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$
2. Compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
3. Initialize a conflict graph $G$

for $r = 5$ to $n$ do

   if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
      delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

   $L \leftarrow$ list of horizon edges visible from $p_r$

   foreach $e \in L$ do

      $f \leftarrow C.\text{create}\_\text{facet}(e, p_r)$; create vtx for $f$ in $G$

      $(f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e)$

      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

   delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$

return $C$
Rand3dConvexHull\((P \subseteq \mathbb{R}^3)\)

pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)

\[ C \leftarrow \text{CH}(P') \]

compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)

initialize conflict graph \(G\)

\[ \text{for } r = 5 \text{ to } n \text{ do} \]

\[ \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ \ p_r \notin C \} \]

delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)

\[ \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \]

\[ \text{foreach } e \in \mathcal{L} \text{ do} \]

\[ f \leftarrow C.\text{createfacet}(e, p_r); \text{create vtx for } f \text{ in } G \]

\[ (f_1, f_2) \leftarrow \text{previously_incident}_C(e) \]

\[ P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \]

\[ \text{return } C \]
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow CH(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in L$ do

$f \leftarrow C$.create_facet($e, p_r$); create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow$ previously_incident$_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow CH(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

    foreach $e \in \mathcal{L}$ do

      $f \leftarrow C.\text{create}\_\text{facet}(e, p_r)$; create vtx for $f$ in $G$

      $(f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e)$

      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

    foreach $p \in P(e)$ do

      if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \not\in C$ \}

    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$

    foreach $e \in \mathcal{L}$ do

        $f \leftarrow C\. create\_facet(e, p_r)$; create vtx for $f$ in $G$

        $(f_1, f_2) \leftarrow \text{previously\_incident}_C(e)$

        $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

    foreach $p \in P(e)$ do

        if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

return $C$
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$. 

initialize conflict graph $G$

for $r = 5$ to $n$ do 

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then 

{ $p_r \not\in C$ }

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in L$ do

$f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

foreach $p \in P(e)$ do

if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

return $C$
Rand3dConvexHull\((P \subset \mathbb{R}^3)\)

pick non-coplanar set \(P' = \{p_1, \ldots, p_4\} \subseteq P\)

\(C \leftarrow \text{CH}(P')\)

compute rand. perm. \((p_5, \ldots, p_n)\) of \(P \setminus P'\)

initialize conflict graph \(G\)

for \(r = 5\) to \(n\) do

\(\text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then } \{ p_r \not\in C \}\)

delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)

\(\mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r\)

foreach \(e \in \mathcal{L}\) do

\(f \leftarrow C.\text{create\_facet}(e, p_r); \text{create vtx for } f \text{ in } G\)

\((f_1, f_2) \leftarrow \text{previously\_incident}_C(e)\)

\(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)

foreach \(p \in P(e)\) do

\(\text{if } f \text{ is visible from } p \text{ then add edge } (p, f) \text{ to } G\)

return \(C\)
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \((p_5, \ldots, p_n)\) of \( P \setminus P' \)

initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do

\[
\text{if} \quad F_{\text{conflict}}(p_r) \neq \emptyset \quad \text{then} \quad \{ p_r \notin C \}
\]

delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)

\( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)

foreach \( e \in \mathcal{L} \) do

\[ f \leftarrow C.\text{create\_facet}(e, p_r); \text{create vtx for } f \text{ in } G \]

\[ (f_1, f_2) \leftarrow \text{previously\_incident}_C(e) \]

\[ P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \]

foreach \( p \in P(e) \) do

\[
\text{if } f \text{ is visible from } p \text{ then add edge } (p, f) \text{ to } G
\]

return \( C \)
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subseteq P \)

\( C \leftarrow CH(P') \)

compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)

initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do
  if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then \( \{ p_r \notin C \} \)
    delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)

  \( \mathcal{L} \leftarrow \) list of horizon edges visible from \( p_r \)

  foreach \( e \in \mathcal{L} \) do
    \( f \leftarrow C.\text{create\_facet}(e, p_r); \) create vtx for \( f \) in \( G \)
    \( (f_1, f_2) \leftarrow \text{previously\_incident}_C(e) \)
    \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)

  foreach \( p \in P(e) \) do
    if \( f \) is visible from \( p \) then add edge \( (p, f) \) to \( G \)

return \( C \)
Rand3dConvexHull\( (P \subset \mathbb{R}^3) \)

pick non-coplanar set \( P' = \{p_1, \ldots, p_4\} \subset P \)

\( C \leftarrow \text{CH}(P') \)

compute rand. perm. \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)

initialize conflict graph \( G \)

for \( r = 5 \) to \( n \) do

if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then \( \{ p_r \notin C \} \)

delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)

\( \mathcal{L} \leftarrow \) list of horizon edges visible from \( p_r \)

foreach \( e \in \mathcal{L} \) do

\( f \leftarrow C.\text{createfacet}(e, p_r); \) create vtx for \( f \) in \( G \)

\( (f_1, f_2) \leftarrow \) previously_incident\( _C(e) \)

\( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)

foreach \( p \in P(e) \) do

if \( f \) is visible from \( p \) then add edge \( (p, f) \) to \( G \)

delete vtc \( \{p_r\} \cup F_{\text{conflict}}(p_r) \) from \( G \)

return \( C \)
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P\setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow \text{list of horizon edges visible from } p_r$

foreach $e \in L$ do

  $f \leftarrow C$.create_facet($e, p_r$); create vtx for $f$ in $G$

  $(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$

  $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

  foreach $p \in P(e)$ do

    if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

  delete vtc $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$

return $C$

Worst-case running time =
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow \text{CH}(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

\begin{align*}
\text{for} \ r = 5 \ \text{to} \ n \ \text{do} & \\
\quad \text{if} \ F_{\text{conflict}}(p_r) \neq \emptyset & \\{ \ p_r \notin C \ \} \\
\quad & \text{delete all facets in} \ F_{\text{conflict}}(p_r) \ \text{from} \ C \\
\quad & \mathcal{L} \leftarrow \text{list of horizon edges visible from} \ p_r \\
\quad \text{foreach} \ e \in \mathcal{L} & \\
\quad & f \leftarrow C.\text{create}\_\text{facet}(e, p_r); \text{create vtx for} \ f \ \text{in} \ G \\
\quad & (f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e) \\
\quad & P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \text{foreach} \ p \in P(e) & \\
\quad & \text{if} \ f \ \text{is visible from} \ p \ \text{then} \ \text{add edge} \ (p, f) \ \text{to} \ G \\
\quad & \text{delete vtc} \ \{p_r\} \cup F_{\text{conflict}}(p_r) \ \text{from} \ G \\
\end{align*}

return $C$

Worst-case running time =
Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick non-coplanar set $P' = \{p_1, \ldots, p_4\} \subseteq P$

$C \leftarrow CH(P')$

compute rand. perm. $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then \{ $p_r \notin C$ \}

delete all facets in $F_{\text{conflict}}(p_r)$ from $C$

$L \leftarrow$ list of horizon edges visible from $p_r$

foreach $e \in L$ do

$f \leftarrow C.\text{create_facet}(e, p_r)$; create vtx for $f$ in $G$

$(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

foreach $p \in P(e)$ do

if $f$ is visible from $p$ then add edge $(p, f)$ to $G$

delete vtx \{ $p_r$ \} $\cup F_{\text{conflict}}(p_r)$ from $G$

return $C$

Worst-case running time = $O(n^3)$
Analysis

Idea: Bound expected *structural change*
Analysis

**Idea:** Bound expected *structural change*, that is, the total number of facets created by the algorithm.
Analysis

Idea: Bound expected structural change, that is, the total \#facets created by the algorithm.

Lemma. The expected \#facets created is at most $6n - 20$. 
Analysis

**Idea:** Bound expected *structural change*, that is, the total \#facets created by the algorithm.

**Lemma.** The expected \#facets created is at most $6n - 20$.

**Proof.** $E[\#\text{facets created}] =$
Analysis

Idea: Bound expected *structural change*, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$
Analysis

**Idea:** Bound expected *structural change*, that is, the total \#facets created by the algorithm.

**Lemma.** The expected \#facets created is at most $6n - 20$.

**Proof.**

\[
E[\text{\#facets created}] = 4 + \sum_{r=5}^{n} E[\text{\#facets incident to } p_r \text{ in } CH(P_r)]
\]
Analysis

Idea: Bound expected *structural change*, that is, the total \#facets created by the algorithm.

Lemma. The expected \#facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$
Analysis

Idea: Bound expected *structural change*, that is, the total \( \# \) facets created by the algorithm.

Lemma. The expected \( \# \) facets created is at most \( 6n - 20 \).

Proof. \[ E[\# \text{facets created}] = 4 + \sum_{r=5}^{n} E[\# \text{facets incident to } p_r \text{ in } CH(P_r)] \]
\[ = 4 + \sum_{r=5}^{n} \deg(p_r, CH(P_r)) \]
Analysis

Idea: Bound expected *structural change*, that is, the total number of facets created by the algorithm.

Lemma. The expected number of facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$

For $r > 4$:
$E[\deg(p_r, \text{CH}(P_r))] =$
Analysis

Idea: Bound expected structural change, that is, the total \#facets created by the algorithm.

Lemma. The expected \#facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_{r} \text{ in } \text{CH}(P_{r})]$

For $r > 4$:

$E[\text{deg}(p_{r}, \text{CH}(P_{r}))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_{i}, \text{CH}(P_{r}))$
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.** $E[\text{#facets created}] = 4 + \sum_{r=5}^{n} E[\text{#facets incident to } p_r \text{ in } \text{CH}(P_r)]$

For $r > 4$:

$$E[\text{deg}(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, \text{CH}(P_r))$$

$$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \text{deg}(p_i)) - 12 \right]$$
Analysis

Idea: Bound expected structural change, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in CH}(P_r)]$

For $r > 4$:

$E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))$

$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right]$
Analysis

Idea: Bound expected structural change, that is, the total # facets created by the algorithm.

Lemma. The expected # facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$

For $r > 4$:

$E[\text{deg}(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \text{deg}(p_i, \text{CH}(P_r))$

$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \text{deg}(p_i)) - 12 \right]$

$\leq 2 \cdot \# \text{ edges of CH}(P_r)$
Analysis

**Idea:** Bound expected structural change, that is, the total \#facets created by the algorithm.

**Lemma.** The expected \#facets created is at most \(6n - 20\).

**Proof.**

\[
E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]
\]

For \(r > 4\):

\[
E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))
\]

\[
\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right] 
\]

\[
2 \cdot \# \text{ edges of } \text{CH}(P_r)
\]

\[
\leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right]
\]
Analysis

**Idea:** Bound expected *structural change*, that is, the total #facets created by the algorithm.

**Lemma.** The expected #facets created is at most $6n - 20$.

**Proof.**

$$E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in } \text{CH}(P_r)]$$

For $r > 4$:

$$E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))$$

$$\leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right]$$

$$\leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right] \leq 6$$
Analysis

Idea: Bound expected structural change, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. $E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in CH}(P_r)] \leq$

For $r > 4$:

$E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r))$

$\leq \frac{1}{r-4} \left[ \left( \sum_{i=1}^{r} \deg(p_i) \right) - 12 \right]

\leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right] \leq 6$
Analysis

Idea: Bound expected structural change, that is, the total #facets created by the algorithm.

Lemma. The expected #facets created is at most $6n - 20$.

Proof. 

\[
E[\#\text{facets created}] = 4 + \sum_{r=5}^{n} E[\#\text{facets incident to } p_r \text{ in CH}(P_r)] \leq \frac{6n}{-20}
\]

For $r > 4$:

\[
E[\deg(p_r, \text{CH}(P_r))] = \frac{1}{r-4} \sum_{i=5}^{r} \deg(p_i, \text{CH}(P_r)) \leq \frac{1}{r-4} \left[ (\sum_{i=1}^{r} \deg(p_i)) - 12 \right] \leq \frac{1}{r-4} \left[ 2 \cdot (3r - 6) - 12 \right] \leq 6
\]
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```plaintext
Rand3dConvexHull($P \subset \mathbb{R}^3$)
pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
$C \leftarrow \text{CH}(P')$
compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
initialize conflict graph $G$
for $r = 5 \text{ to } n$ do
    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
        delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
        $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
        foreach $e \in \mathcal{L}$ do
            $f \leftarrow C.\text{createfacet}(e, p_r)$; create vtx for $f$ in $G$
            $(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$
            $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
            foreach $p \in P(e)$ do
                if $f$ visible from $p$ then add edge $(p, f)$ to $G$
        delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
    return $C$
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```plaintext
Rand3dConvexHull(P ⊂ \mathbb{R}^3)

- pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
- $C \leftarrow \text{CH}(P')$
- compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
- initialize conflict graph $G$

for $r = 5$ to $n$ do
  if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
    delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
    $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
    foreach $e \in \mathcal{L}$ do
      $f \leftarrow C$\_create$\text{facet}(e, p_r)$; create vtx for $f$ in $G$
      $(f_1, f_2) \leftarrow \text{previously}_\text{incident}_C(e)$
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ do
        if $f$ visible from $p$ then add edge $(p, f)$ to $G$
    delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
  return $C$
```
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```
Rand3dConvexHull(P ⊂ R^3)
    { O(n) time }
    pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
    $C \leftarrow CH(P')$
    compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
    initialize conflict graph $G$
    for $r = 5$ to $n$ do
        if $F_{conflict}(p_r) \neq \emptyset$ then
            delete all facets in $F_{conflict}(p_r)$ from $C$
            $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
            foreach $e \in \mathcal{L}$ do
                $f \leftarrow C$.create_facet($e, p_r$); create vtx for $f$ in $G$
                $(f_1, f_2) \leftarrow$ previously_incident$_C(e)$
                $P(e) \leftarrow P_{conflict}(f_1) \cup P_{conflict}(f_2)$
                foreach $p \in P(e)$ do
                    if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{conflict}(p_r)$ from $G$
    return $C$
```


Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop)

```
Rand3dConvexHull(P ⊂ R^3)

{ O(n) time }

pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts
$C \leftarrow CH(P')$
compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$
initialize conflict graph $G$

for $r = 5$ to $n$ do
    if $F_{\text{conflict}}(p_r) \neq \emptyset$ then
        delete all facets in $F_{\text{conflict}}(p_r)$ from $C$
        $\mathcal{L} \leftarrow$ list of horizon edges visible from $p_r$
        foreach $e \in \mathcal{L}$ do
            $f \leftarrow C$.create_facet$(e, p_r)$; create vtx for $f$ in $G$
            $(f_1, f_2) \leftarrow \text{previously_incident}_C(e)$
            $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
            foreach $p \in P(e)$ do
                if $f$ visible from $p$ then add edge $(p, f)$ to $G$
            delete vtx $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$
        return $C$
```
**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) \begin{cases} 
\text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
C \leftarrow \text{CH}(P') \\
\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad \quad f \leftarrow C.\text{create_facet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
\quad \quad \quad (f_1, f_2) \leftarrow \text{previously_incident}_C(e) \\
\quad \quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \quad \quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\end{cases}
\]

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{\text{conflict}}(p_r)|) = \)

\( O(n) \) time
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) = \begin{cases} 
\text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad \quad f \leftarrow C.\text{create facet}(e, p_r); \text{create vtx for } f \text{ in } G \\
\quad \quad \quad (f_1, f_2) \leftarrow \text{previously incident}_C(e) \\
\quad \quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \quad \quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \quad \quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\text{return } C 
\end{cases}
\]

Stage \( r \) of for-loop \( (w/o \text{ outer foreach loop}) \)
takes time \( O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r) \)

This part of for-loop in total:
Running Time

**Theorem:** The convex hull of a set of \(n\) pts in \(\mathbb{R}^3\) can be computed in \(O(n \log n)\) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) \begin{cases}
\text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad f \leftarrow C.\text{create}\_\text{facet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
\quad \quad (f_1, f_2) \leftarrow \text{previously}\_\text{incident}_C(e) \\
\quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\end{cases}
\]

Stage \(r\) of for-loop (w/o outer foreach loop) takes time \(O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r)\)

This part of for-loop in total: \(E[\#\text{facets deleted}] = \)
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r)$

This part of for-loop in total: $E[#\text{facets deleted}] = \leq E[#\text{facets created}] = $
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

```
Rand3dConvexHull(P ⊂ \mathbb{R}^3)
    pick set \( P' = \{p_1, \ldots, p_4\} \subseteq P \) of 4 non-coplanar pts
    \( C \leftarrow \text{CH}(P') \)
    compute a random permutation \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)
    initialize conflict graph \( G \)
    for \( r = 5 \) to \( n \) do
        if \( F_{\text{conflict}}(p_r) \neq \emptyset \) then
            delete all facets in \( F_{\text{conflict}}(p_r) \) from \( C \)
            \( \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \)
            foreach \( e \in \mathcal{L} \) do
                \( f \leftarrow \text{C.createfacet}(e, p_r); \text{create vtx for } f \) in \( G \)
                \( (f_1, f_2) \leftarrow \text{previously_incident}_C(e) \)
                \( P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \)
                foreach \( p \in P(e) \) do
                    if \( f \) visible from \( p \) then add edge \( (p, f) \) to \( G \)
                delete vtc \( \{p_r\} \cup F_{\text{conflict}}(p_r) \) from \( G \)
        return \( C \)
```

Stage \( r \) of for-loop \( \text{(w/o outer foreach loop)} \)
takes time \( O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r) \)
This part of for-loop in total:
\[
E[#\text{facets deleted}] = E[#\text{facets created}] \leq \text{Lemma}
\]
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

$E[\#\text{facets deleted}] = 
\leq E[\#\text{facets created}] = O(n).$ 

**Lemma**
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total: $E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.

Outer foreach-loop:
Running Time

Theorem: The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

Stage $r$ of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)$

This part of for-loop in total: $E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n)$.

Outer foreach-loop: in stage $r$: $O(\sum_{e \in \mathcal{L}} |P(e)|)$

Rand3dConvexHull($P \subset \mathbb{R}^3$)

pick set $P' = \{p_1, \ldots, p_4\} \subseteq P$ of 4 non-coplanar pts

compute a random permutation $(p_5, \ldots, p_n)$ of $P \setminus P'$

initialize conflict graph $G$

for $r = 5$ to $n$ do

if $F_{\text{conflict}}(p_r) \neq \emptyset$ then

delte all facets in $F_{\text{conflict}}(p_r)$ from $C$

let $\mathcal{L}$ be a list of horizon edges visible from $p_r$

foreach $e \in \mathcal{L}$ do

$\ D$; create a facet $e$, $p_r$; create a vertex for $e$ in $G$

$(f_1, f_2) \leftarrow \text{previously incident}_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

foreach $p \in P(e)$ do

if $f$ visible from $p$ then add edge $(p, f)$ to $G$

delete vertex $\{p_r\} \cup F_{\text{conflict}}(p_r)$ from $G$

return $C$
Running Time

Theorem: The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r) \)

This part of for-loop in total: \( E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n) \).

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
- in total:
The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3) =
\begin{align*}
\text{pick set } P' &= \{p_1, \ldots , p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
C &\leftarrow \text{CH}(P') \\
\text{compute a random permutation } (p_5, \ldots , p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad f \leftarrow C.\text{createfacet}(e, p_r); \text{ create vtx for } f \text{ in } G \\
\quad \quad (f_1, f_2) \leftarrow \text{previously\_incident}_{C}(e) \\
\quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\text{return } C
\end{align*}
\]

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r) \).

This part of for-loop in total: \( E[\#\text{facets deleted}] = \leq E[\#\text{facets created}] = O(n) \).

Outer foreach-loop:
\begin{itemize}
\item in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
\item in total: \( O\left( \sum_{e \text{ on horizon at some moment}} |P(e)| \right) \)
\end{itemize}
Running Time

Theorem: The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

\[
\text{Rand3dConvexHull}(P \subset \mathbb{R}^3)
\begin{align*}
\text{pick set } P' = \{p_1, \ldots, p_4\} \subseteq P \text{ of 4 non-coplanar pts} \\
C \leftarrow \text{CH}(P') \\
\text{compute a random permutation } (p_5, \ldots, p_n) \text{ of } P \setminus P' \\
\text{initialize conflict graph } G \\
\text{for } r = 5 \text{ to } n \text{ do} \\
\quad \text{if } F_{\text{conflict}}(p_r) \neq \emptyset \text{ then} \\
\quad \quad \text{delete all facets in } F_{\text{conflict}}(p_r) \text{ from } C \\
\quad \quad \mathcal{L} \leftarrow \text{list of horizon edges visible from } p_r \\
\quad \text{foreach } e \in \mathcal{L} \text{ do} \\
\quad \quad f \leftarrow \text{C.createfacet}(e, p_r); \text{create vtx for } f \text{ in } G \\
\quad \quad (f_1, f_2) \leftarrow \text{previously Incident}_C(e) \\
\quad \quad P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2) \\
\quad \text{foreach } p \in P(e) \text{ do} \\
\quad \quad \quad \text{if } f \text{ visible from } p \text{ then add edge } (p, f) \text{ to } G \\
\quad \quad \text{delete vtx } \{p_r\} \cup F_{\text{conflict}}(p_r) \text{ from } G \\
\text{return } C
\end{align*}
\]

Stage \( r \) of for-loop (w/o outer foreach loop) takes time
\[
O(|F_{\text{conflict}}(p_r)|) = O(\#\text{facets deleted when adding } p_r)
\]

This part of for-loop in total: \( E[\#\text{facets deleted}] = O(\#\text{facets created}] = O(n). \)

Lemma

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e \in \mathcal{L}} |P(e)|) \)
- in total:
\[
O \left( \sum_{e \text{ on horizon at some moment}} |P(e)| \right) = O(n \log n)
\]
Running Time

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

```plaintext
Rand3dConvexHull(P \subset \mathbb{R}^3)
pick set \( P' = \{p_1, \ldots, p_4\} \subseteq P \) of 4 non-coplanar pts
\( C \leftarrow CH(P') \)
compute a random permutation \( (p_5, \ldots, p_n) \) of \( P \setminus P' \)
initialize conflict graph \( G \)
for \( r = 5 \) to \( n \) do
  if \( F_{conflict}(p_r) \neq \emptyset \) then
    delete all facets in \( F_{conflict}(p_r) \) from \( C \)
    \( \mathcal{L} \leftarrow \) list of horizon edges visible from \( p_r \)
    foreach \( e \in \mathcal{L} \) do
      \( f \leftarrow C.createfacet(e, p_r) \); create vtx for \( f \) in \( G \)
      \( (f_1, f_2) \leftarrow \) previously_incident\( _C(e) \)
      \( P(e) \leftarrow P_{conflict}(f_1) \cup P_{conflict}(f_2) \)
      foreach \( p \in P(e) \) do
        if \( f \) visible from \( p \) then add edge \((p, f)\) to \( G \)
      delete vtc \( \{p_r\} \cup F_{conflict}(p_r) \) from \( G \)
  return \( C \)
```

Stage \( r \) of for-loop (w/o outer foreach loop) takes time \( O(|F_{conflict}(p_r)|) = O(#facets deleted when adding \( p_r \)) \)

This part of for-loop in total:
\[
E[#facets deleted] = E[#facets created] = O(n). \tag{Lemma}
\]

Outer foreach-loop:
- in stage \( r \): \( O(\sum_{e\in \mathcal{L}} |P(e)|) \)
- in total:
\[
O\left(\sum_{e \text{ on horizon at some moment}} |P(e)|\right) = O(n \log n)
\]
Running Time

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.

```plaintext
Rand3dConvexHull(\(P \subseteq \mathbb{R}^3\))

pick set \(P' = \{p_1, \ldots, p_4\} \subseteq P\) of 4 non-coplanar pts
compute a random permutation \((p_5, \ldots, p_n)\) of \(P \setminus P'\)
initialize conflict graph \(G\)

for \(r = 5\) to \(n\) do
    if \(F_{\text{conflict}}(p_r) \neq \emptyset\) then
        delete all facets in \(F_{\text{conflict}}(p_r)\) from \(C\)
    \(\mathcal{L} \leftarrow\) list of horizon edges visible from \(p_r\)
    foreach \(e \in \mathcal{L}\) do
        \(f \leftarrow C.\text{create}\_\text{facet}(e, p_r)\); create vtx for \(f\) in \(G\)
        \((f_1, f_2) \leftarrow\) previously incident \(C(e)\)
        \(P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)\)
    foreach \(p \in P(e)\) do
        if \(f\) visible from \(p\) then add edge \((p, f)\) to \(G\)

return \(C\)
```

Stage \(r\) of for-loop (w/o outer foreach loop) takes time $O(|F_{\text{conflict}}(p_r)|) = O(#\text{facets deleted when adding } p_r)$

This part of for-loop in total:

\[E[#\text{facets deleted}] \leq E[#\text{facets created}] = O(n).\] (Lemma)

Outer foreach-loop:
- in stage \(r\): $O(\sum_{e \in \mathcal{L}} |P(e)|)$
- in total:

\[O\left(\sum_{e \text{ on horizon at some moment}} |P(e)|\right) = O(n \log n)\]
Running times – expected vs. worst case

**Theorem:** The convex hull of a set of $n$ pts in $\mathbb{R}^3$ can be computed in $O(n \log n)$ expected time.
Running times – expected vs. worst case

**Theorem:** The convex hull of a set of \( n \) pts in \( \mathbb{R}^3 \) can be computed in \( O(n \log n) \) expected time.

**Exercise:** Give a simple deterministic algorithm that computes the convex hull in \( O(n^2) \) (worst-case) time.
Convex Hulls and Half-Space Intersections
Convex Hulls and Half-Space Intersections

Plane

Plane
Convex Hulls and Half-Space Intersections

Plane

Define duality $\star$ between pts and (non-vertical) lines:
Convex Hulls and Half-Space Intersections

Define duality \( \star \) between pts and (non-vertical) lines:

For \( p = (p_x, p_y) \),
Convex Hulls and Half-Space Intersections

Define duality \( \ast \) between pts and (non-vertical) lines:

For \( p = (p_x, p_y) \),

\[ p \]

\[ \text{primal} \]
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$. 
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$. 

\[ p \]

 primal

 dual
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$,
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$,
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$. 
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$. 
Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.
Define duality $\star$ between pts and (non-vertical) lines: 

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$. 

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$. 
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe:
Convex Hulls and Half-Space Intersections

Plane

Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line. $\star$ is incidence-preserving:
Convex Hulls and Half-Space Intersections

Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line. $\ast$ is incidence-preserving: $p \in \ell \iff \ell^\ast \in p^\ast$.
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving:
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^* : y = p_x x - p_y$.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

* is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

* is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

*$\ast$ is incidence-preserving: $p \in \ell \iff \ell^\ast \in p^\ast$

*$\ast$ is order-preserving: $p$ above $\ell \iff \ell^\ast$ above $p^\ast$
Define duality $\ast$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\ast: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\ast$ to be the pt $q$ with $q^\ast = \ell$, that is, $\ell^\ast = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\ast$ is incidence-preserving: $p \in \ell \iff \ell^\ast \in p^\ast$

$\ast$ is order-preserving: $p$ above $\ell \iff \ell^\ast$ above $p^\ast$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star$: $y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
* is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$
* is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

**Observe:** Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^* \in p^*$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^*$ above $p^*$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star : y = p_x x - p_y$.

For $\ell : y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$,
that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.
$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$
$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Plane

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$,
that is, $\ell^\star = (m, -b)$.

Observe: Let $p \in \mathbb{R}^2$ and let $\ell$ be a non-vertical line.

$\star$ is incidence-preserving: $p \in \ell \iff \ell^\star \in p^\star$

$\star$ is order-preserving: $p$ above $\ell \iff \ell^\star$ above $p^\star$
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^*: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^*$ to be the pt $q$ with $q^* = \ell$, that is, $\ell^* = (m, -b)$.

Observe: Upper convex hulls of pts $\leftrightarrow$ lower envelopes of lines
Convex Hulls and Half-Space Intersections

Define duality $\star$ between pts and (non-vertical) lines:

For $p = (p_x, p_y)$, define the line $p^\star: y = p_x x - p_y$.

For $\ell: y = mx + b$, define $\ell^\star$ to be the pt $q$ with $q^\star = \ell$, that is, $\ell^\star = (m, -b)$.

Observe:
- upper convex hulls of pts $\leftrightarrow$ lower envelopes of lines
- can compute intersections of “lower/upper” half planes (spaces) via upper/lower convex hulls
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the *unit paraboloid* in $\mathbb{R}^3$. 
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the *unit paraboloid* in $\mathbb{R}^3$. 
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 

$p = (p_x, p_y, 0)$
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$. 

$p = (p_x, p_y, 0)$
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
p' = (p_x, p_y, p_x^2 + p_y^2)
\]

\[
p = (p_x, p_y, 0)
\]
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]

\[
p' = (p_x, p_y, p_x^2 + p_y^2)
\]

\[
p = (p_x, p_y, 0)
\]
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

Note that $p' \in h(p)$.
Voronoi Diagrams Revisited

Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$$h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$$

Note that $p' \in h(p)$.

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$
Let \( U : z = x^2 + y^2 \) be the \textit{unit paraboloid} in \( \mathbb{R}^3 \).

Note that \( p' \in h(p) \).
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$p' = (p_x, p_y, p_x^2 + p_y^2)$

$p = (p_x, p_y, 0)$

$|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[ h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).

\[ p' = (p_x, p_y, p_x^2 + p_y^2) \]

\[ |pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \]
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[ h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$\text{h}(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in \text{h}(p)$.
Let $U: z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$. 

$p = (p_x, p_y, 0)$

$q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$

$|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[ h(p): z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).

\[ q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2)) \]

\[ |pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \]
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$p = (p_x, p_y, 0)$

$q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$

$|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$

$z_{p'} - z_{q(p)} = |pq|^2$
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

Let \( p = (p_x, p_y, 0) \) and \( p' = (p_x, p_y, p_x^2 + p_y^2) \).

\[ h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).

\[ q(p) = (q_x, q_y, 2p_xq_x + 2p_yq_y - (p_x^2 + p_y^2)) \]

\[ |pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \]

\[ |pq|^2 = (pq - zq(p)) \]

\[ q' = (q_x, q_y, 2p_xq_x + 2p_yq_y - (p_x^2 + p_y^2)) \]

\[ \Rightarrow h(p) \text{ and } U \text{ encode dist. betw. } p \text{ and any other pt in } z = 0. \]
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)
\]

Note that \( p' \in h(p) \).

\[
q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))
\]

\[
\Rightarrow h(p) \text{ and } U \text{ encode dist. betw. } p \text{ and any other pt in } z = 0.
\]

\[
\Rightarrow h(p) \cap U = \{p'\}
\]
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the *unit paraboloid* in $\mathbb{R}^3$. 

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$

$|pq| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$

$z_{q'} - z_{q(p)} = |pq|^2$

$\Rightarrow h(p)$ and $U$ encode dist. betw. $p$ and any other pt in $z = 0$. 

$\Rightarrow h(p) \cap U = \{p'\} \Rightarrow h(p)$ is tangent to $U$ (in $p'$)
Voronoi Diagrams Revisited

Let $U : z = x^2 + y^2$ be the *unit paraboloid* in $\mathbb{R}^3$.

Let $U : z = x^2 + y^2$ be the unit paraboloid in $\mathbb{R}^3$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2)$

Note that $p' \in h(p)$.

$h(p) \cap U = \{p'\} \Rightarrow h(p)$ is tangent to $U$ (in $p'$)
Voronoi Diagrams Revisited

Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[ h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \]

Note that \( p' \in h(p) \).

\[ q(p) = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2)) \]

\[ |pq|^2 = (p_x - q_x)^2 + (p_y - q_y)^2 \]

\[ \Rightarrow h(p) \) and \( U \) encode dist. betw. \( p \) and any other pt in \( z = 0 \).

\[ \Rightarrow h(p) \cap U = \{p'\} \Rightarrow h(p) \) is tangent to \( U \) (in \( p' \) )
Let \( U : z = x^2 + y^2 \) be the unit paraboloid in \( \mathbb{R}^3 \).

\[
\begin{align*}
\text{Let } U & : z = x^2 + y^2 \text{ be the unit paraboloid in } \mathbb{R}^3. \\
h(p) : z = (2p_x)x + (2p_y)y - (p_x^2 + p_y^2) \\
\text{Note that } p' \in h(p).
\end{align*}
\]

\[
\begin{align*}
h(q) & = h(p) \cap U = \{p'\} \Rightarrow h(p) \text{ is tangent to } U \text{ (in } p')
\end{align*}
\]
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{ h(p) \mid p \in P \}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. 
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$.
The Upper Envelope Strikes Back

**Theorem:** Let \( P \subset \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{ h(p) \mid p \in P \} \).
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. 
The Upper Envelope Strikes Back

**Theorem:** Let \( P \subset \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{ h(p) \mid p \in P \} \). Let \( \mathcal{E}(\mathcal{H}) \) be the upper envelope of \( \mathcal{H} \).
Theorem: Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. 
The Upper Envelope Strikes Back

**Theorem:** Let \( P \subseteq \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{h(p) \mid p \in P\} \).

Let \( \mathcal{E}(\mathcal{H}) \) be the upper envelope of \( \mathcal{H} \).

The projection of \( \mathcal{E}(\mathcal{H}) \) on \( z = 0 \) is
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$. 

\[ \text{Vor}(P) \]

\[ \text{Vor}(P) \]
**Theorem:** Let \( P \subset \mathbb{R}^2 \times \{0\} \) and \( \mathcal{H} = \{ h(p) \mid p \in P \} \).

Let \( \mathcal{E}(\mathcal{H}) \) be the upper envelope of \( \mathcal{H} \).

The projection of \( \mathcal{E}(\mathcal{H}) \) on \( z = 0 \) is \( \text{Vor}(P) \).

\[ \text{can compute } \text{Vor}(P) \text{ in } \mathbb{R}^2 \text{ via upper envelope in } \mathbb{R}^3 \]
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$.

Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$.

The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

**Exercise 11.10**

can compute $\text{Vor}(P)$ in $\mathbb{R}^2$
via upper envelope in $\mathbb{R}^3$

\[ \text{Vor}(P) \]
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

- Can compute $\text{Vor}(P)$ in $\mathbb{R}^2$ via upper envelope in $\mathbb{R}^3$
- Upper envelope in $\mathbb{R}^3$ is in one-to-one correspondence to lower convex hull of the pt set $\mathcal{H}^*$

Exercise 11.10
**The Upper Envelope Strikes Back**

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$. Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$. The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

- can compute $\text{Vor}(P)$ in $\mathbb{R}^2$ via upper envelope in $\mathbb{R}^3$
- exercise 11.10

upper envelope in $\mathbb{R}^3$ is in one-to-one correspondence to lower convex hull of the pt set $\mathcal{H}^*$
The Upper Envelope Strikes Back

**Theorem:** Let $P \subset \mathbb{R}^2 \times \{0\}$ and $\mathcal{H} = \{h(p) \mid p \in P\}$.

Let $\mathcal{E}(\mathcal{H})$ be the upper envelope of $\mathcal{H}$.

The projection of $\mathcal{E}(\mathcal{H})$ on $z = 0$ is $\text{Vor}(P)$.

- Can compute $\text{Vor}(P)$ in $\mathbb{R}^2$ via upper envelope in $\mathbb{R}^3$
- Exercise 11.10
- Upper envelope in $\mathbb{R}^3$ is in one-to-one correspondence to lower convex hull of the pt set $\mathcal{H}^*$
- Use algorithm Rand3dConvexHull!