CHAN'S ALGORITHM

• easier than ultimate?

• modify Jarvis march to work on $\left\lceil \frac{n}{m} \right\rceil$ groups of $m$ points

  0) Compute C.H. of every group : $\frac{n}{m} \cdot O(m \log m)$
  1) Start w/ $X_{\text{max}}$
  2) Compute tangent to every group : $\frac{n}{m} \cdot O(\log m)$
  3) Pick steepest : first hull edge : $\frac{n}{m}$
  4) Repeat w/ new point $\rightarrow h$ times

  total $O(n \log m) + h \cdot \frac{n}{m} \cdot O(\log m)$

$m = 5$
Jarvis hull for \( \frac{n}{m} \leq m \cdot \text{gons} \) : total: \( O(n \log m) + h \cdot \frac{n}{m} \cdot O(\log m) \)

What should \( m \) be?
\[
\begin{align*}
    m = n & \rightarrow h \cdot \log n + n \cdot \log n \\
    m = \sqrt{n} & \rightarrow h \cdot \sqrt{n} \cdot \log n + n \cdot \log n \\
    m = \log n & \rightarrow h \cdot n \cdot \frac{\log \log n}{\log n} + n \cdot \log \log n \\
    m = 1 & \rightarrow h \cdot n \; \text{(regular Jarvis)} \\
    m = h & \rightarrow n \cdot \log h + n \cdot \log h \; \text{PERFECT}
\end{align*}
\]

But we don't know \( h \). Guess! Binary search?

First try: \( \frac{n}{2} \ldots O(n \log n) \)

Can't afford to try anything much larger than \( h \). (\( h^c \) is ok)
Critical twist: For any guess $m$, when you try Jarvis, STOP after $m$ hull edges
(you already realize your guess was wrong)

$\frac{n}{m} \cdot \frac{n}{m} \log m$

[construct $\frac{n}{m}$ group hulls]

$+ m \cdot \frac{n}{m} \log m$

FIND 1 tangent
FIND 1 hull edge
one attempt
(abandon if incomplete)

$O(n \log m)$ per guess

better than $O(n \log h)$
for $m \leq h$
• We are willing to FAIL many times, as long as it doesn't cost too much & we approach h quickly.

• We can't guess too high.

**Sequence of guesses:** \( m = 2^{2^t} \) for \( t = 1, 2, \ldots \) while \( m < n \)

• How many times will we overshoot? **ONCE**

**Guess cost:** \( O(n \log m) = n \cdot \log 2^{2^t} = n \cdot 2^t \)

\[ \begin{align*}
2, & \quad 4, \quad 8, \quad 16 \\
\sum, & \quad < x
\end{align*} \]

**Perfect guess:** \( h = 2^{2^t} + 1 \) \( \text{cost} = x = n \cdot 2^t \)

**Worst overshoot:** \( 2^{2^t+1} \) \( \text{cost} = n \cdot 2^{t+1} = 2x \)

\( \frac{1}{4} T(\text{all work}) = T(\text{last fail}) < T(\text{perfect guess}) \leq T(\text{overshoot = success}) = \frac{1}{2} T(\text{all work}) \)
• We are willing to FAIL many times, as long as it doesn't cost too much & we approach h quickly.
• We can't guess too high.

**Sequence of guesses:** \( m = 2^t \) for \( t = 1, 2, \ldots \) while \( m < n \)

• How many times will we overshoot? **ONCE**

**Guess cost:** \( O(n \log m) = n \cdot \log 2^t = n \cdot 2^t \)

\[
\begin{align*}
n & \rightarrow n.2 & n.4 & n.8 & n.16 \\
\leq & \rightarrow < x
\end{align*}
\]

- Perfect guess: \( h = 2^{t+1} \)
- Worst overshoot: \( 2^{t+1} \)

\[
\sum_{t=1}^{\log \log h} n2^t = O(n \cdot 2^{\frac{\log \log h}{\log \log \log h}}) = O(n \log h)
\]

Example: last bad guess \( \frac{\log \log h}{\log \log \log h} < 2x \) (when \( t = \log \log h \))

\( m = h \)
A slightly incorrect example

\[ h = 16 \]
\[ n = 43 \]
$t=1 \rightarrow m=2$

I'm using $m=2^t$.

The example shows how to quit after guessing too low, but we are not incrementing fast enough to get a good run time.
10 quadrilaterals
1 triangle

$t=2 \rightarrow m=4$

This is actually the first iteration, with $t=1$, $m=2^t$, for a real $t$. 
$t = 2 \rightarrow m = 4$
\( t=2 \rightarrow m=4 \)
5 groups of 8
1 triple

\[ t=3 \rightarrow m=8 \]
5 groups of 8

1 triple

\( t = 3 \rightarrow m = 8 \)
5 groups of 8

1 triple

\[ t = 3 \rightarrow m = 8 \]
5 groups of 8
1 triple

$t=3 \rightarrow m=8$
5 groups of 8

1 triple

$t=3 \rightarrow m=8$
5 groups of 8

1 triple

\( t = 3 \rightarrow m = 8 \)
5 groups of 8
1 triple

\( t=3 \rightarrow m=8 \)
5 groups of 8

1 triple

t=3  →  m=8
5 groups of 8
1 triple

\[ t = 3 \implies m = 8 \]
2 groups of 16
1 group 11

\[ t = 4 \rightarrow m = 16 \]

Real iteration 2
\[ t = 2, m = 2^t \]
2 groups of 16
1 group 11

\( t=4 \rightarrow m=16 \)
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

\[ t=4 \rightarrow m=16 \]
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

\( t = 4 \rightarrow m = 16 \)
2 groups of 16
1 group 11

\( t = 4 \rightarrow m = 16 \)
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

$t=4 \rightarrow m=16$
2 groups of 16
1 group 11

\( t = 4 \rightarrow m = 16 \)

DONE

NOTE that if \( 9 < k < 15 \) \( m = 16 \) would have worked just fine
Why didn’t I draw an actual example?

Guess = $2^t$: 4, 16, 256, 65536, 4294967296 5th guess

I said it’s ok to over guess $h_c$.

Overshoot guess = $2^{2^{t+1}}$

Last failed guess = $2^{2^t}$

\[
\text{ratio: } 2^{t+1} / 2^t = 2^t
\]

\[
\text{i.e. } 2^{2^{t+1}} = 2^{2^t} \cdot 2^{2^t}
\]

overguess = (failed guess)$^2$

< $h^2$