UNSW CPMSoc | CITADEL | Securities

MEET & GREET

+ O-WEEK CONTEST DEBRIEF

KEITH BURROWS THEATRE

WED, FEB 16, 6 - 8 PM

PRIZE PRESENTATION, FREE PIZZA, CITADEL SECURITIES MERCH GIVEAWAY, AND MUCH MORE!

Contest site and tasks (participation not required):
contest.unswcpmsoc.com

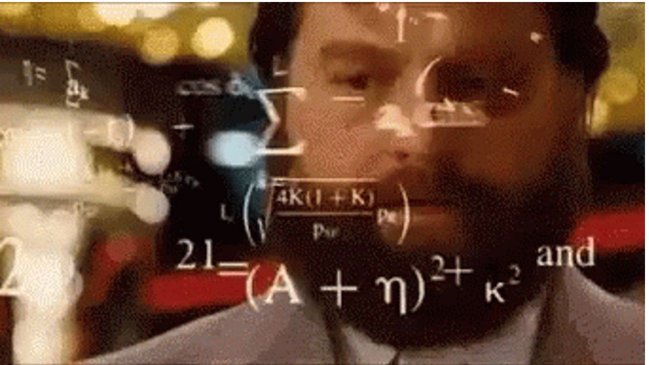
Plan for tonight

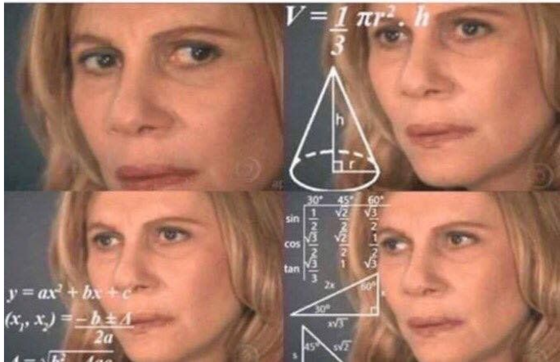
6:10-6:30	Introduction, some solutions
6:30-6:45	???
6:45-7:15	More solutions
7:15-7:30	Prize announcement and conclusion
7:30-?	Merch giving, Pizza eating, and Conversation having

All Task Statements:
https://www.unswcpmsoc.com/assets/OWeek22Tasks.pdf

UNSW CPMSoc

O-Week Contest 2022 Debrief





Some Sum

91 viewers, 73 solvers, 98 submissions

some sum

/sʌm/ /sʌm/

$$1 + 10 + 100 + 1000 + 1000 - 100 + 10 + 1$$

People solving Some Sum
then leaving, with expected
prize money of about 67 cents

Turning Point

61 half solves, 53 full solves, 106+101 submissions

Turning Point (Part 1)

Find a non-constant polynomial in x which:

- Has real integer coefficients, and
- Has a stationary point at $x = 3$.

Turning Point (Part 2)

Find a non-constant polynomial in x which:

- Has real integer coefficients, and
- Has a stationary point at x = sqrt(3).

Sandwich

4 partial solves, 29 full solves, 107 submissions

Sandwich

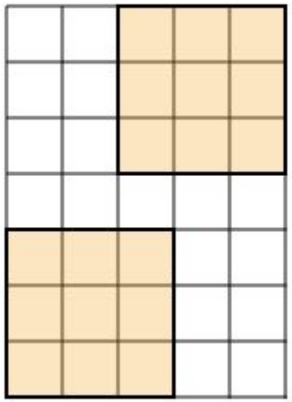
You have a Wcm by Hcm slice of bread where W and H are both integers, and you would like to make a sandwich. To do so, you will cut out two equal sized squares from the bread, also with integer side length. Assuming you can only cut out squares with edges parallel to the edges of the original slice, write a program which given W and H, calculates the minimum amount of bread left over at the end.

- W, H <= 100
- The area of bread is at least 2

Sandwich

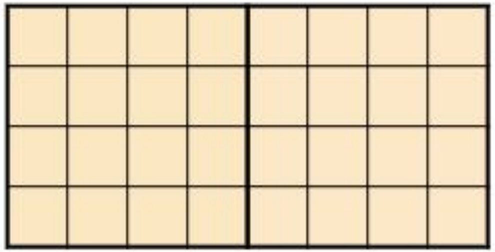
E.g. W = 5, H = 7

Output: 17



W = 8, H = 4

Output: 0



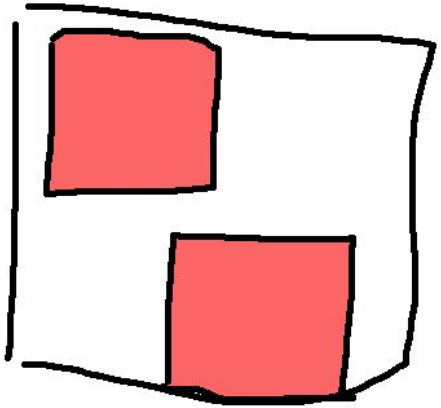
Sandwich

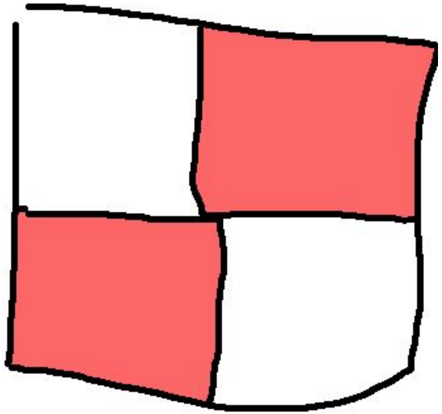
Observation 1: Minimising the left-over bread is equivalent to maximising the bread used.

Sandwich

Observation 1: Minimising the left-over bread is equivalent to maximising the bread used.

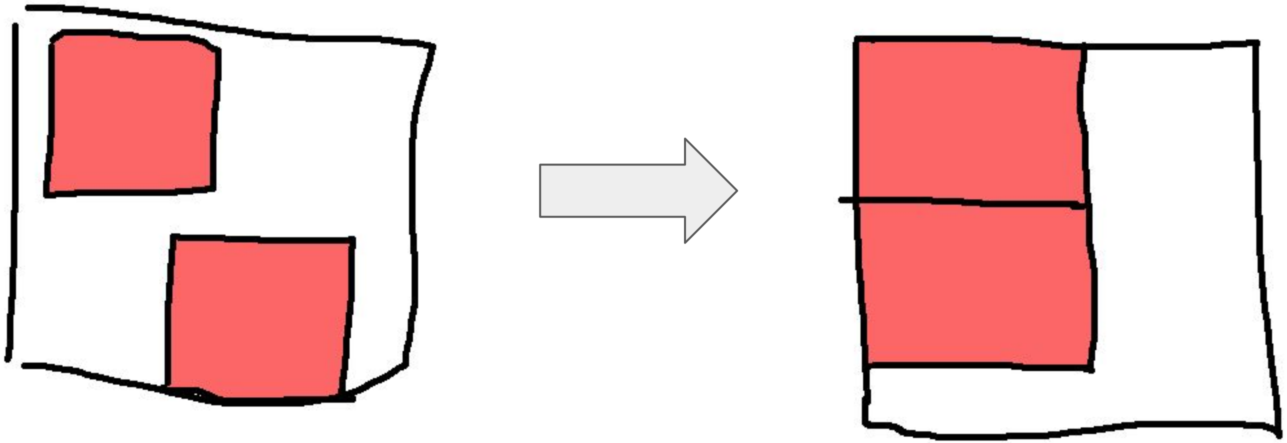
- But what strategy should we use to arrange the bread?





Sandwich

Observation 2: We only care about the case where both squares are stacked together into the top left corner



Sandwich

The new problem: Find the largest s by 2s rectangle that will fit within a WxH rectangle.

Sandwich

The new problem: Find the largest s by 2s rectangle that will fit within a WxH rectangle.

- W and H are both smaller than 100, so the maximum size of s must also be 100
- Loop through every possible value of s, and find the largest value of s where s <= W and 2s <= 2H
- Make sure to repeat for the 2s by s rectangle too!

Tennis

35 half solves, 12 full solves, 129+34 submissions

Tennis

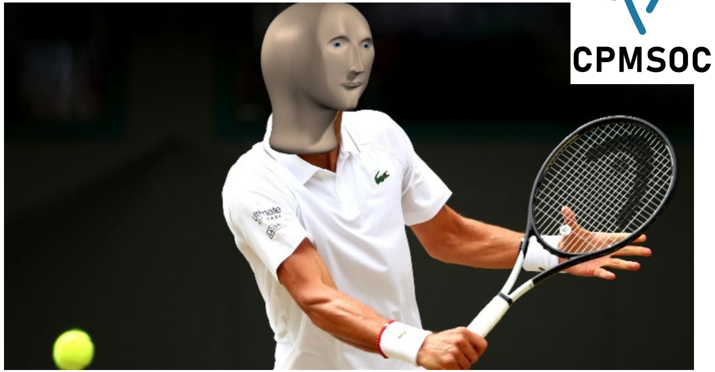
Tennis scoring rules:

- First person to win 6 games wins the set
- Games end when:
    - One player has won 4 points, AND
    - The difference between the players’ scores is 2 or more
    - The higher-scoring player wins the game

Objectives:

1. Win the set
2. Win the minimum proportion out of all points in the set

Tennis



First person to win 6 games wins the set

We want to win the set

We win exactly 6 games, and our opponent wins up to 5

Games end when one player scores more than 4 and score difference is ≥ 2

Games can end 0:4, 1:4 or (2+k):(4+k), where k≥0.

Tennis

Win 6, lose up to 5

Games can end 0:4, 1:4 or (2+k):(4+k), where k≥0.

Fun exercise: Prove that each game we win should have the same final score, and each game we lose should have the same final score.

Tennis

Final score vector

$= [\text{our total}, \text{their total}] = 6 * W + a * L$

Where:

- $0 \le a \le 5$
- $W = [4,0], [4,1]$ or $[4+n,2+n]$ for $n \ge 0$
  - $= [4,0]$ plus either $[0,0]$, $[0,1]$ or $[n,2+n]$
  - $= [4,0]$ plus either $[0,0]$, $[0,1]$ or $[0,2]+[n,n]$
- $L = [0,4], [1,4]$ or $[2+m,4+m]$ for $m \ge 0$
  - $= [0,4]$ plus either $[0,0]$, $[1,0]$ or $[2+m,m]$
  - $= [0,4]$ plus either $[0,0]$, $[1,0]$ or $[2,0]+[m,m]$

Total is

$6 * [4,0] + 6 * ([0,0],[0,1] \text{ or } [0,2]+[n,n]) + k * ([0,0],[1,0] \text{ or } [2,0]+[m,m]) + k * [0,4]$

i.e.

$6 * [4,0] + 6 * ([0,0],[0,1] \text{ or } [0,2]) + k * ([0,0],[1,0] \text{ or } [2,0]) + k * [0,4] + [T,T]$

CPMSOC

Arc CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

Tennis

Total is 6 * [4,0] + 6 * ([0,0],[0,1] or [0,2]) + k * ([0,0],[1,0] or [2,0]) + [T,T]

So our final score is $[24, 4k] + [T, T] + [y, x]$ where x is between 0 and 12 inclusive and y is between 0 and $2k$ inclusive.

The proportion of total points we have won is

$$\frac{24 + T + y}{24 + T + y + 4k + T + x},$$

where numerator and denominator are integers.

Weakest Link

5 partial solves, 26 full solves, 122 submissions

Weakest Link

Given an array of length N, for every element a_i, write a program to find the new minimum element in the array when a_i is removed.

2 <= N <= 100,000, and 1 <= a_i <= 10^9.

Our program must run within 1 second.

E.g. If the input was: [1, 2, 3]

Then the final output would be: [2, 1, 1]

Weakest Link

Solution 1: Brute Force

For each of the N elements, make a new copy of the array with that element missing. Then for each of the N arrays, work out the minimum element and print it out.

Complexity is O(N^2)

Weakest Link

Solution 1: Brute Force

For each of the N elements, make a new copy of the array with that element missing. Then for each of the N arrays, work out the minimum element and print it out.

Complexity is O(N^2)

- **But N can be up to 10^5, which means N^2 is 10^10!**
- **A fast computer can only handle about 10^8 computations per second**
- **Hence our solution is too slow for the 1 second time limit.**
- **It is, however, good enough to pick up 5 out of 10 of the marks for this question.**

Weakest Link

Observation: The minimum element in the array after removal can only ever be either the smallest number, or the second smallest number.

Weakest Link

Observation: The minimum element in the array after removal can only ever be either the smallest number, or the second smallest number.

Solution 2:

1. **Go through the array and find the smallest number**
2. **Go through the array a second time, ignoring the smallest number, to find the second smallest number**
3. **Go through the array one last time:**
    a. **If the number is the smallest number, then print out the second smallest number**
    b. **Otherwise, just print out the smallest number**

This algorithm makes 3 passes of the array, so the complexity is O(3n) = O(n).

Squid Game

Solves: Part 1: 25, Part 2: 22, Part 3: 8
126+70+61 submissions

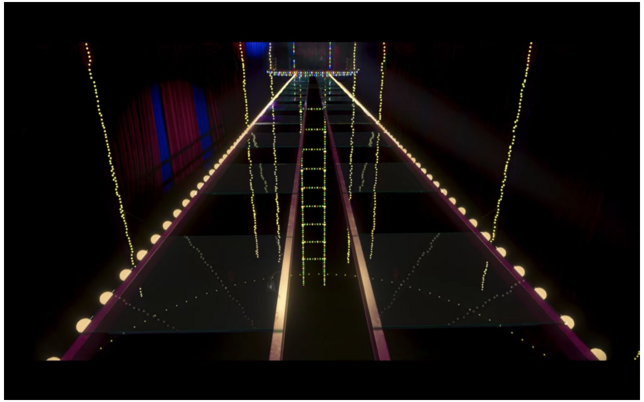
Squid Game (Part 1): $N = 7, P = 3$

There is a game with N turns and P players.

On each turn, either 0 or 1 player is eliminated, with equal probability.

Game ends after N turns or once all players are eliminated, whichever comes first.

Expected number of survivors?



Squid Game (Part 1): N = 7, P = 3

There is a game with N turns and P players.

On each turn, either 0 or 1 player is eliminated, with equal probability.

Game ends after N turns or once all players are eliminated, whichever comes first.

Expected number of winners?

P \ N	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0	0
1	1							
2	2							
3	3							

Squid Game (Part 1): N = 7, P = 3

CPMSOC

There is a game with N turns and P players.

On each turn, either 0 or 1 player is eliminated, with equal probability.

Game ends after N turns or once all players are eliminated, whichever comes first.

Expected number of winners?

P \ N	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128
2	2	3/2	1	5/8	3/8	7/32	1/8	9/64
3	3	5/2	2	3/2	17/16	23/32	15/32	19/64

Squid Game (Part 2): $N = 10^6$, $P = 10^6$

P \ N	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128
2	2	3/2	1	5/8	3/8	7/32	1/8	9/64
3	3	5/2	2	3/2	17/16	23/32	15/32	19/64

CPMSOC

Arc CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

Squid Game (Part 2): $N = 10^6$, $P = 10^6$

Let E(N, P) be expected winners with N turns and P players.

By inspection, f(N, N) = N/2 gg (proof hint: symmetry between x and N-x winners).

(more generally, f(N, P) = P - N/2, for N <= P)

P \ N	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128
2	2	3/2	1	5/8	3/8	7/32	1/8	9/64
3	3	5/2	2	3/2	17/16	23/32	15/32	19/64

Squid Game (Part 3): N=10^{18}, P=5×10^{17}, answer to 3sf

Let E(N, P) be expected number of winners with N turns and P players.

Consider the probabilities that exactly r players win for all r:

$$
\begin{aligned}
E(N,P) &= \sum xP(x) \\
&= \frac{1}{2^N}\sum_{r=0}^{P} r\cdot\binom{N}{P-r} \\
&= \frac{1}{2^N}\sum_{r=0}^{P} (P-r)\cdot\binom{N}{r}
\end{aligned}
$$

Squid Game (Part 3): $N=10^{18}$, $P=5\times10^{17}$, answer to 3sf

$$
\begin{aligned}
f(2P,P) &= \frac{1}{2^{2P}}\sum_{r=0}^{P}(P-r)\binom{2P}{r} \\
&= \cdots \ (\text{exercise to the reader :P}) \\
&= \frac{(2P-1)!}{(P-1)!2^{2P}}
\end{aligned}
$$

(or to Wolfram Alpha)

Squid Game (Part 3): N=10^{18}, P=5×10^{17}, answer to 3sf

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

$$\begin{aligned}
&= \frac{P^2}{2P}\cdot\frac{1}{2^{2P}}\cdot\frac{(2P)!}{P!^2} \\
&\approx \frac{P}{2}\cdot\frac{1}{2^{2P}}\cdot\frac{\sqrt{4\pi P}\left(\frac{2P}{e}\right)^{2P}}{2\pi P\left(\frac{P}{e}\right)^{2P}} \text{ (from Stirlings approximation)} \\
&\approx \cdots \\
&\approx \frac{\sqrt{2P}}{\sqrt{8\pi}} \\
f(10^{18}, 5\times10^{17}) &\approx \frac{10^9}{\sqrt{8\pi}}
\end{aligned}$$

Squid Game (Part 4??): General case, answer to 3sf

Squid Game (Part 4??): General case, answer to 3sf

P = N/2

P = N/2 + sqrt(N)

From normal approximation,

$$\frac{1}{2^N}\binom{N}{k} \approx \int_{k-1/2}^{k+1/2} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)\right) dx,$$

where $\mu = N/2$ and $\sigma = \sqrt{N}/2$.

Thus, assuming $N \geq P$ and stuff is large, the answer is

$$\begin{aligned}
\frac{1}{2^N}\sum_{k=0}^{P}(P-k)\binom{N}{k} &\approx \int_{-1/2}^{P+1/2} (P-x)\sqrt{\frac{2}{\pi N}} \exp\left(-\frac{1}{2N}(2x-N)^2\right) dx. \\
&= \cdots \\
&= \left[\frac{1}{4}\sqrt{\frac{2N}{\pi}} \exp\left(-\frac{1}{2N}(2x-N)^2\right)\right]_{-1/2}^{P+1/2} \\
&\quad - (P-N/2)\left(\Phi\left(\frac{2P+1-N}{\sqrt{N}}\right) - \Phi\left(\frac{-1-N}{\sqrt{N}}\right)\right),
\end{aligned}$$

where Φ is the CDF of the standard normal distribution.

Interesting stuff only happens close to $P = N/2$, precisely at this value it is slightly degenerate.

Advertising

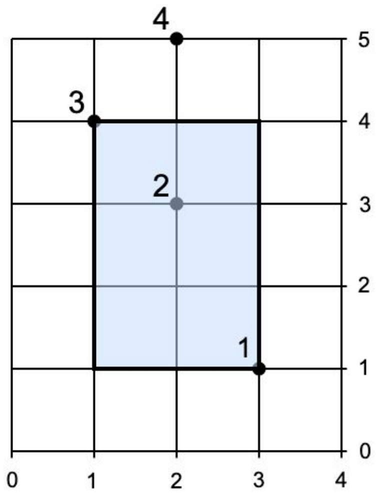
9 partial solves, 2 full solves, 135 submissions
Solution Credits: Technocoder

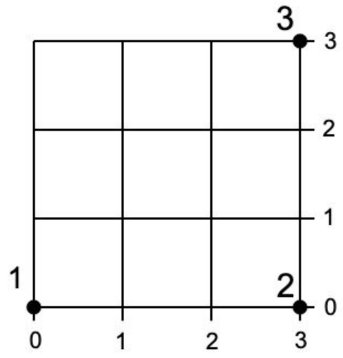
Advertising

Given N points in the plane, find the largest rectangle such that

- Its top left corner is one of those points, and
- Its bottom right corner is one of those points

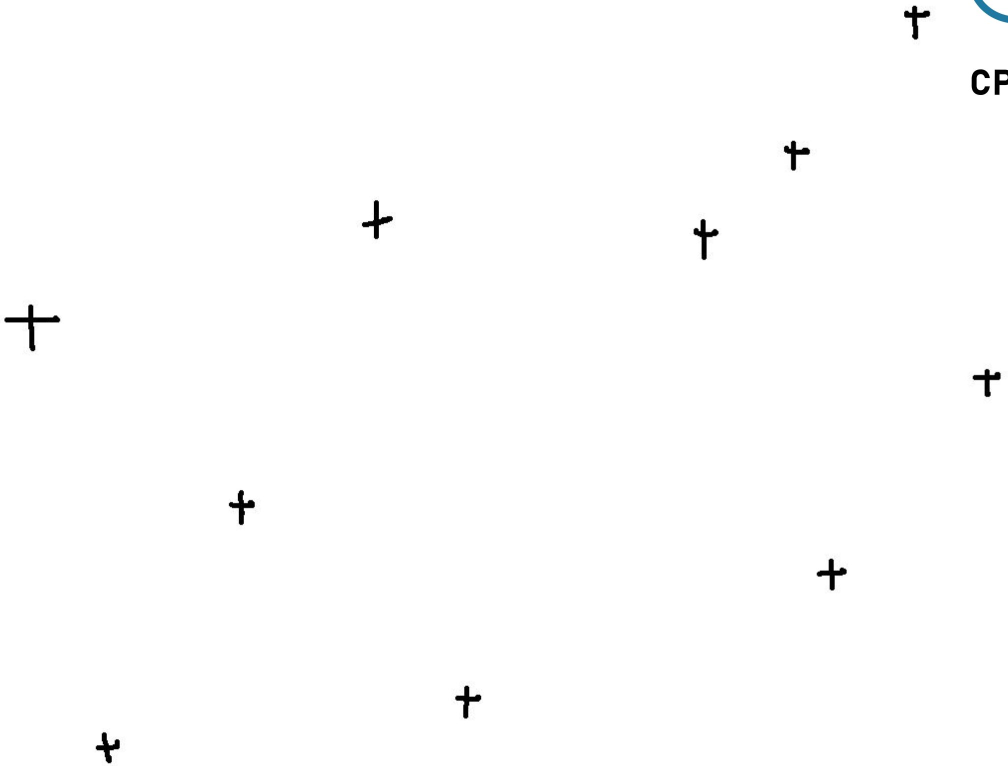
(Trivial if N is small)





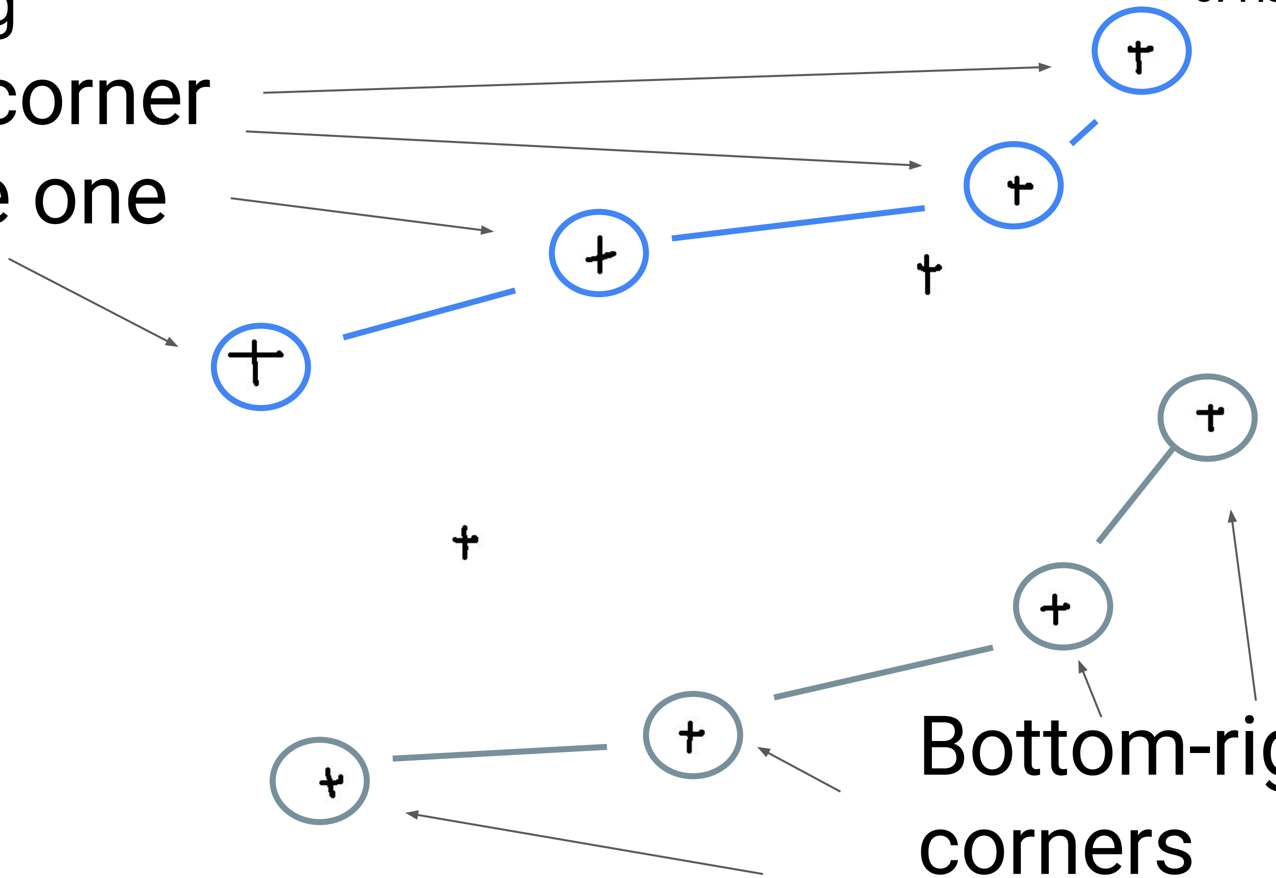
Advertising

Given some points...



Advertising

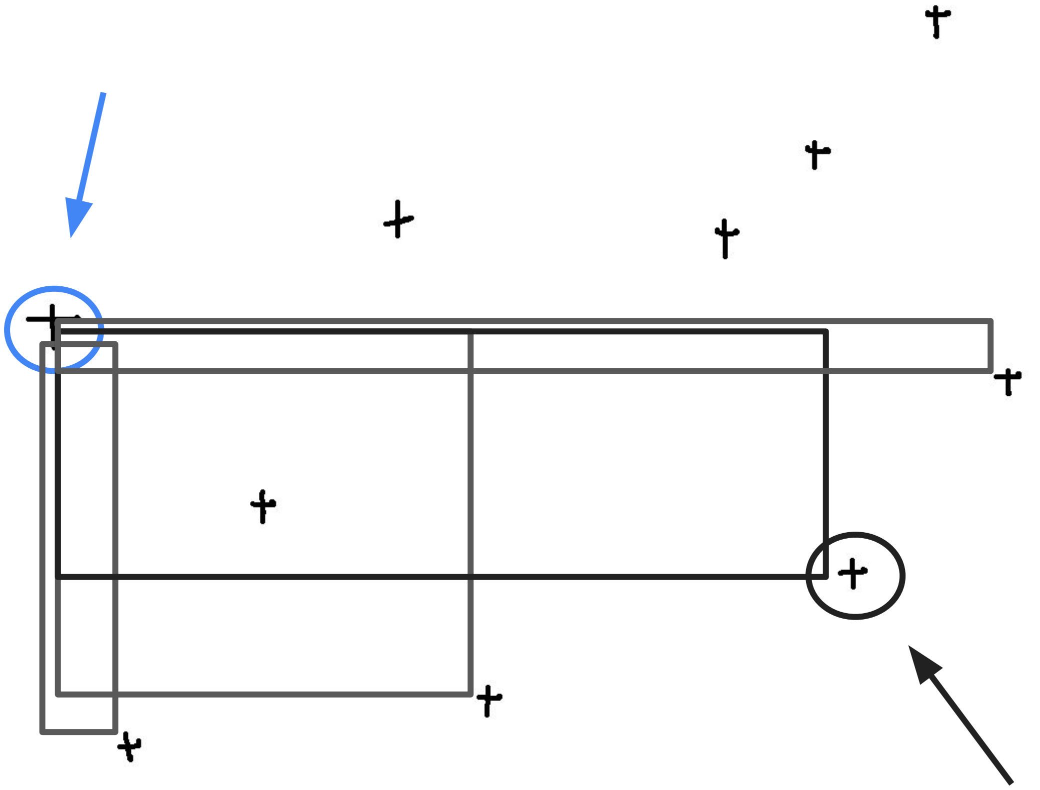
Top-left corner has to be one of these



Bottom-right corners

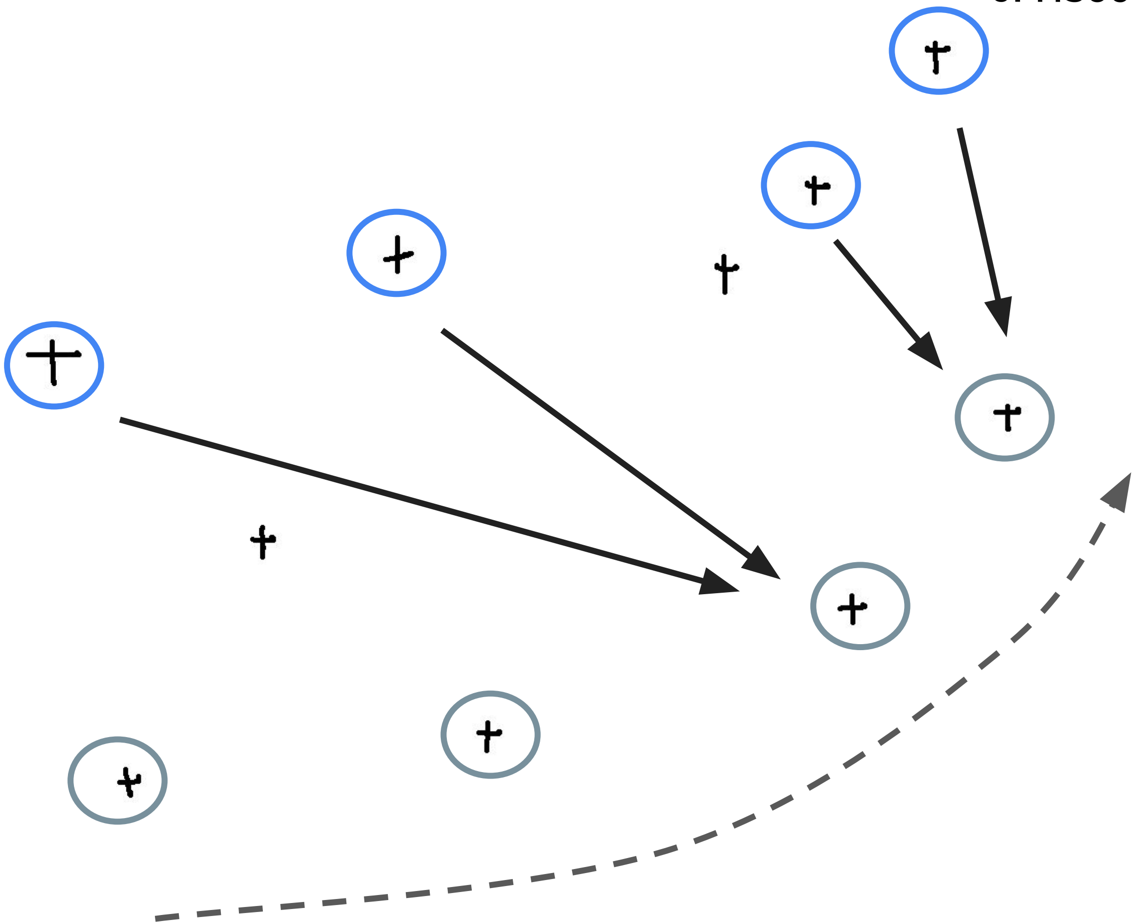
Advertising

Every top-left corner has an optimal bottom-right corner (biggest rectangle)



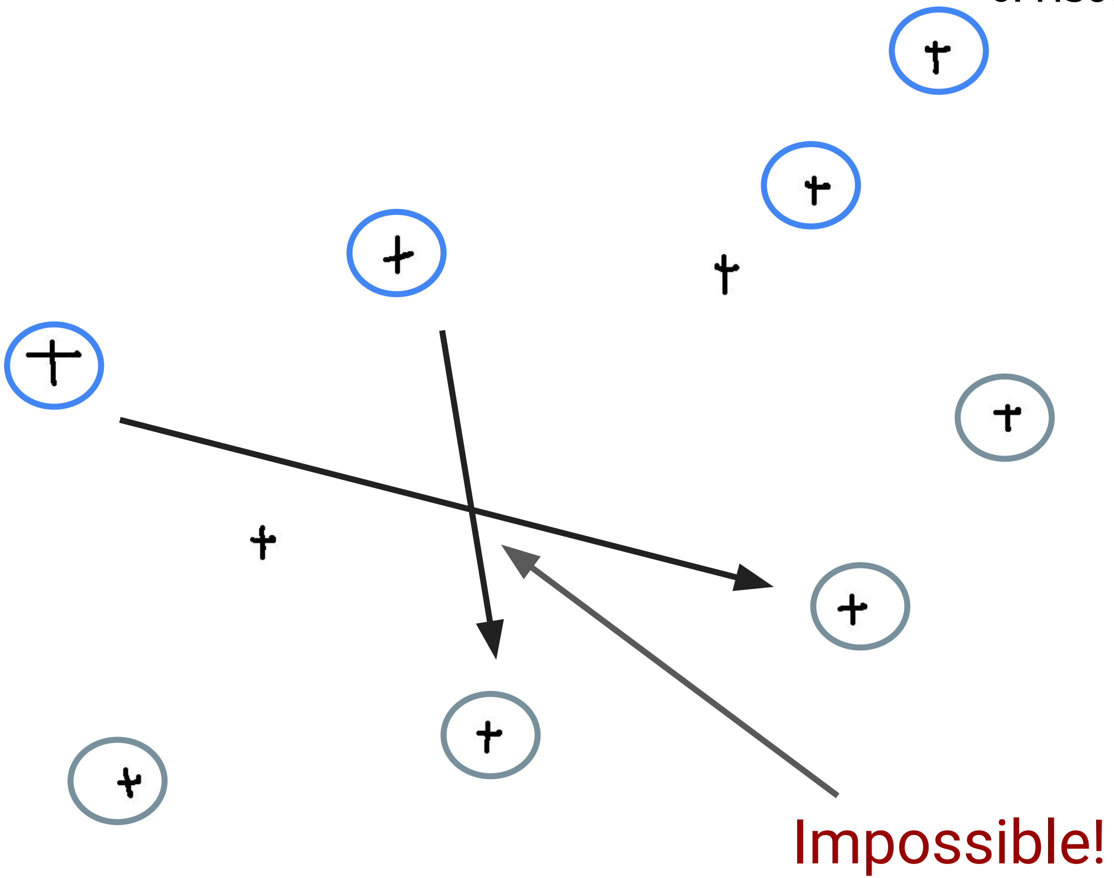
Advertising

The optimal bottom-right corners are monotonic (go from left to right)



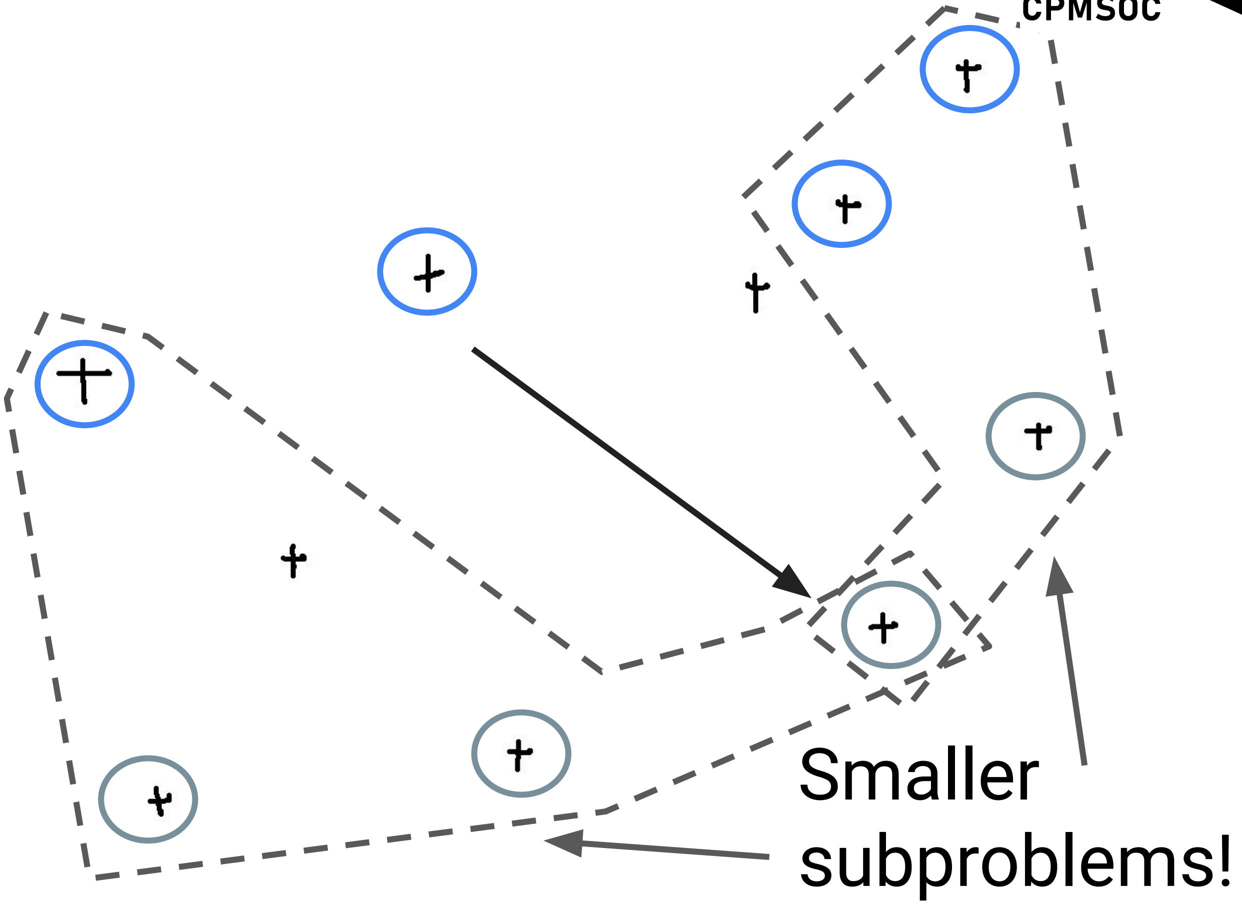
Advertising

This can't ever happen! (proof not shown here)



Advertising

Solution:
Divide and conquer to find all the optimal points then pick the best pair



Complexity:
$O(n \log n)$

Cooked Sum

7 partial solves, 2 full solves, 13 submissions
Solution Credits: ryno & Bill

Cooked Sum

Find a simple expression in n equivalent to

$$\sum_{k=1}^{n-1} \binom{n}{k} k^{k-1} (n-k)^{n-k-1}.$$

And prove it.

Cooked Sum (ryno)

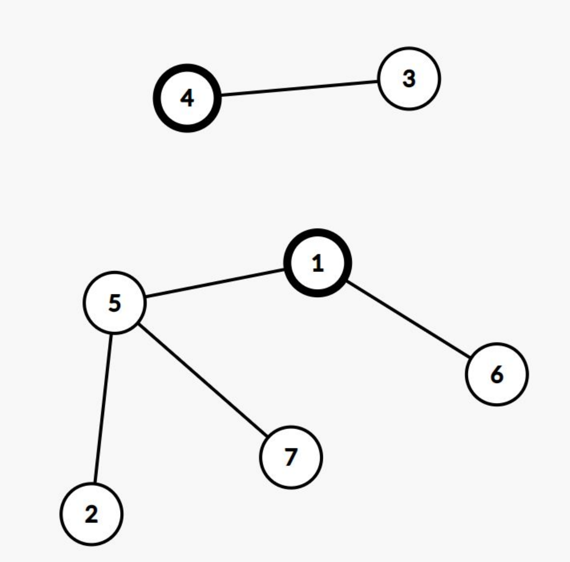
$$2n^{n-2}(n-1) = \sum_{k=1}^{n-1} \binom{n}{k} k^{k-1}(n-k)^{n-k-1}$$

$$2n^{n-2}(n-1) = \sum_{k=1}^{n-1} \binom{n}{k} \cdot k^{k-2} \cdot (n-k)^{n-k-2} \cdot k(n-k)$$

n = number of nodes

n-1 = number of edges

n^{n-2} = number of possible trees



Cooked Sum (ryno)

$$n^{n-2} = \frac{1}{2(n-1)} \sum_{k=1}^{n-1} \binom{n}{k} \cdot k^{k-2} \cdot (n-k)^{n-k-2} \cdot k(n-k)$$

say n = 7, k = 2

This means there are 7 nodes, grouped into trees of size 2 and 7-2

Step 1: Allocate numbers, say {3, 4} and {1, 2, 5, 6, 7}

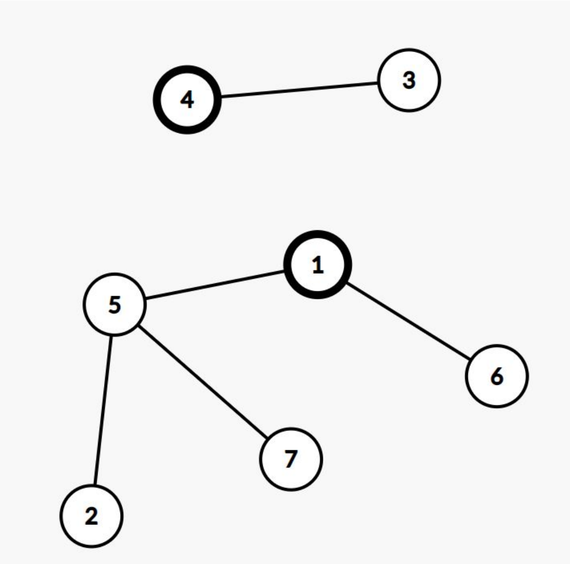
Step 2: Find combinations for tree of size 2

Step 3: Find combinations for tree of size 5

Step 4: Edge to connect trees

Double counting:

- each tree counted n-1 times
- then counted twice when considering n = 7, k = 2



CPMSOC

ARC CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

Cooked Sum (Parking Function Edition)

`2, 12, 96, 1000, 12960, 201684, 3670016, 76527504, 1800000000`

~~Brute force~~

~~Induction~~

Cooked Sum (Bill)
Double counting 1

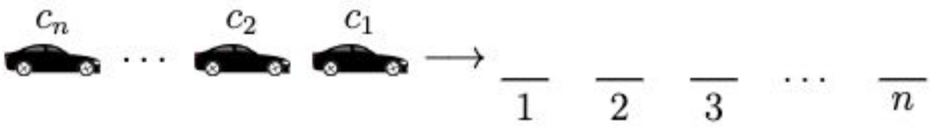
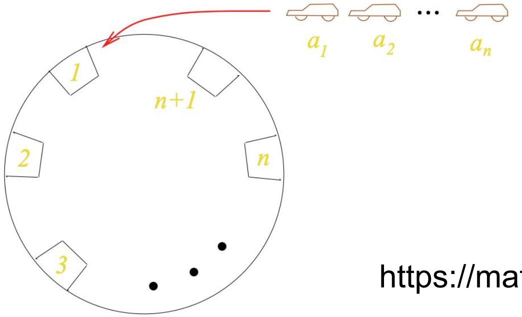


Figure 1: Parking function illustration.

Theorem 3.2 (Closed Form). *The number of parking functions of length n is given by*

$$P(n) = (n+1)^{n-1}$$



https://math.mit.edu/~rstan/transparencies/parking.pdf https://www.unswcpmsoc.com/assets/BillCookedSum.pdf

Cooked Sum (Bill)
Double counting 1

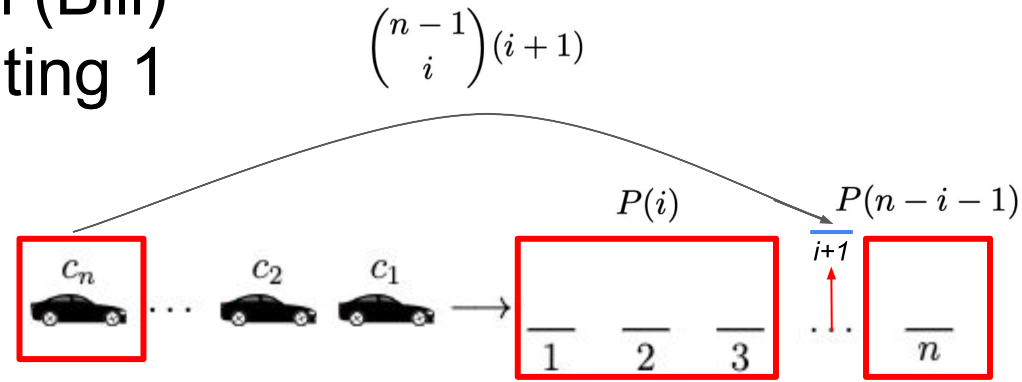


Figure 1: Parking function illustration.

Theorem 3.3 (Recurrence Form). *The number of parking functions of length n is given by*

$$P(n) = \sum_{i=0}^{n-1} \binom{n-1}{i} (i+1) P(i) P(n-i-1)$$

https://www.unswcpmsoc.com/assets/BillCookedSum.pdf

Cooked Sum (Bill)
Equating the two

We can therefore equate the closed form and the recurrence form for use in this problem

$$P(n-1) = n^{n-2} = \sum_{i=0}^{n-2} \binom{n-2}{k} (k+1)P(k)P((n-k-2))$$

Looking back

$$2(n-1)\boxed{n^{n-2}} = \sum_{k=1}^{n-1} \binom{n}{k} \boxed{k^{k-1}}\boxed{(n-k)^{n-k-1}}$$

Similar terms!

End of presentation