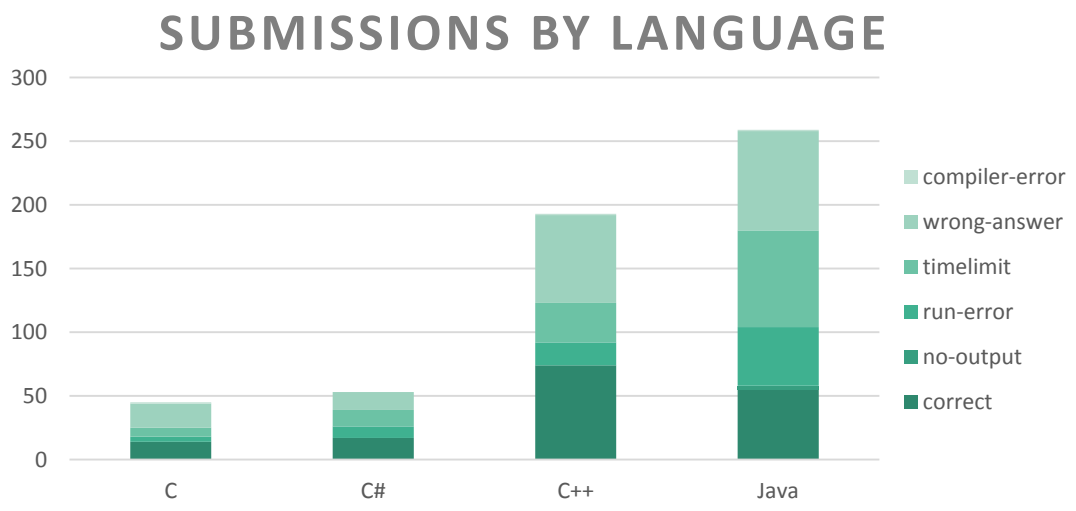
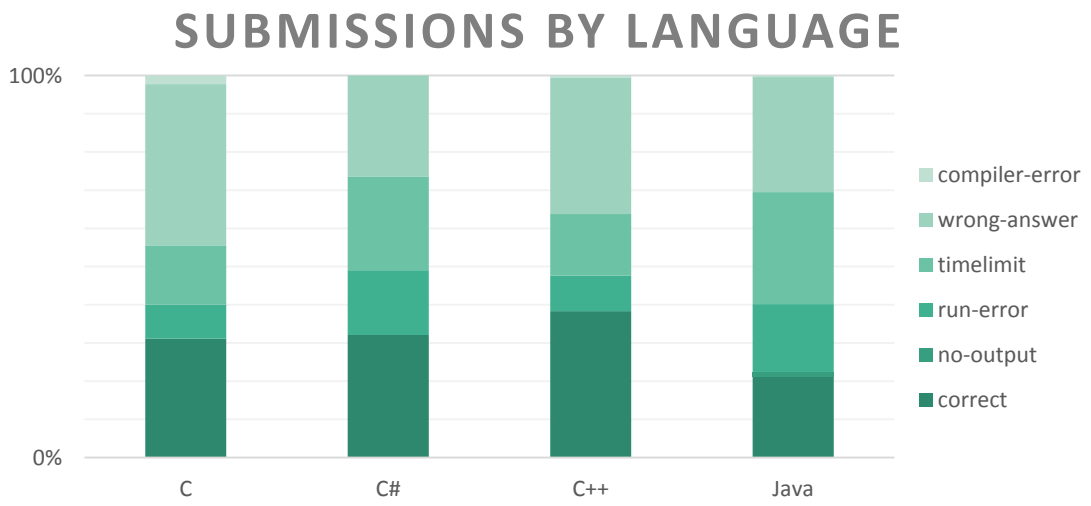
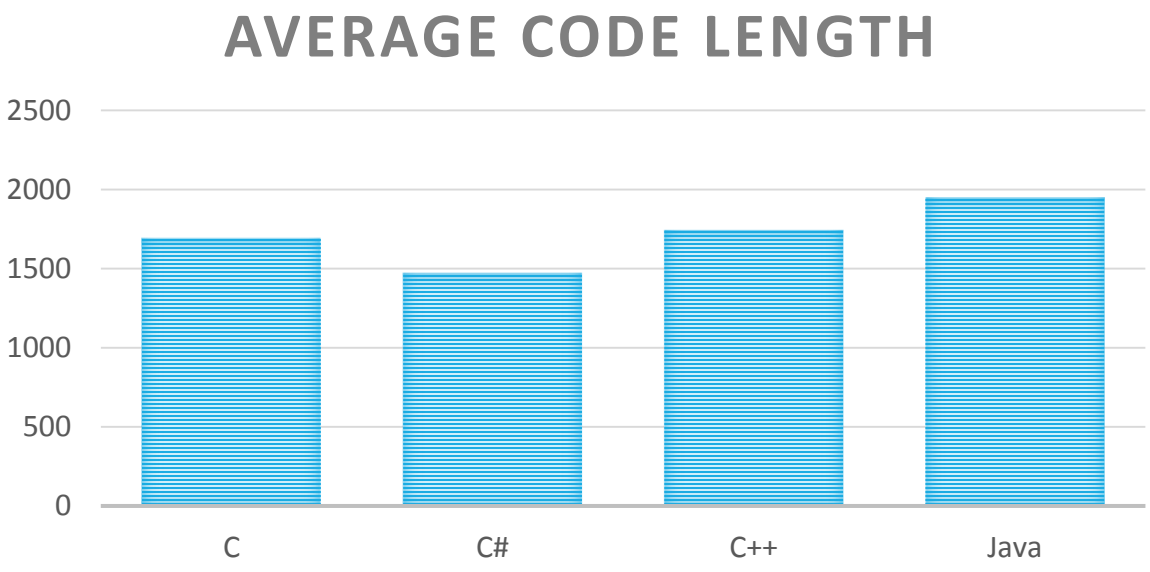
BAPC 2012

STATS + SOLUTIONS + SCORES

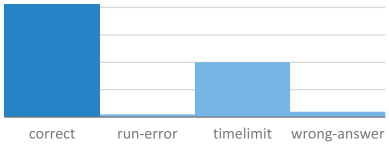
SUBMISSIONS OVER TIME

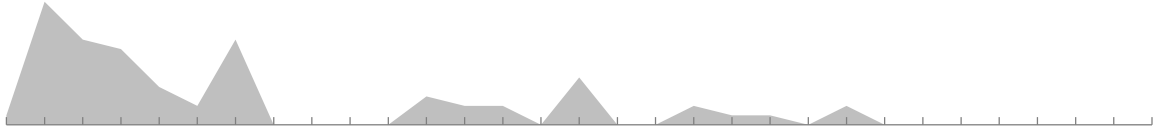

SUBMISSIONS BY LANGUAGE
300
250
200
150
100
50
0
C
C#
C++
Java
compiler-error
wrong-answer
timelimit
run-error
no-output
correct


SUBMISSIONS BY LANGUAGE
100%
0%
C
C#
C++
Java
compiler-error
wrong-answer
timelimit
run-error
no-output
correct


AVERAGE CODE LENGTH
2500
2000
1500
1000
500
0
C
C#
C++
Java

Solutions





Encoded Message

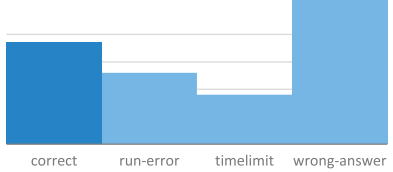
R	o	s	e	s
A	r	e	R	e
d	V	i	o	l
e	t	s	A	r
e	B	l	u	e

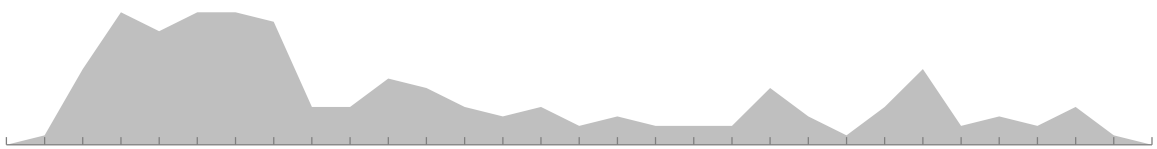
$\Rightarrow$

e	e	d	A	R
B	t	V	r	o
l	s	i	e	s
u	A	o	R	e
e	r	l	e	s

```
int m = sqrt(message.length)
for i = 0 to m-1
    for j = 0 to m-1
        write(message[m-i-1+j*m])
writeline()
```

$O(n)$





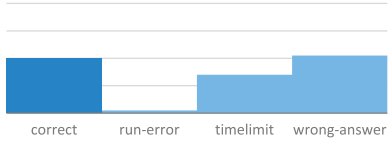
Integer Lists

Actually reversing the lists is too slow

Use a **Deque** (double ended queue)

- Or keep 2 pointers to the begin and end yourself

$O(n)$



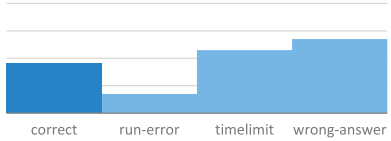
Good Coalition

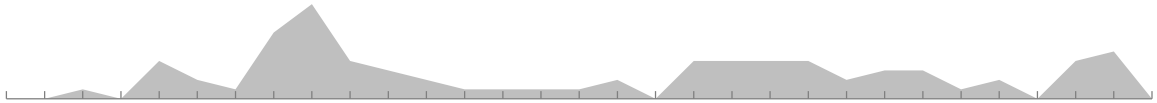
Variant on **Knapsack**

Use **dynamic programming**

- $dp(i, j) =$ maximum chance using exactly j seats considering only the first i parties
- Loop over parties and over number of seats to update this state
- Find maximum chance considering all parties for 76 .. 150 seats

$O(n \cdot 150)$





Fire

First, do a **Breadth First Search** for the fires

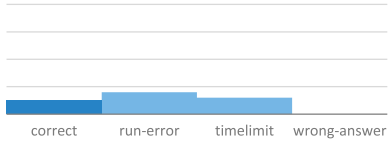
- Start simultaneously from all fires
- For each floor, compute the time it will burn (distance to closest fire)

Then do another BFS to find the shortest path to the edge – without burning

Note that there might not be a fire

- And even then it might be impossible!

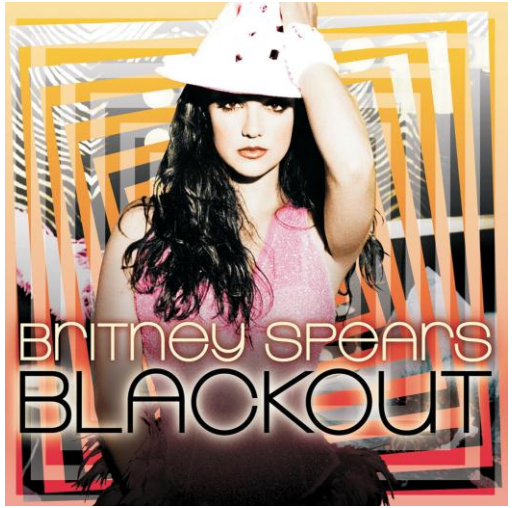
$O(w \cdot h)$

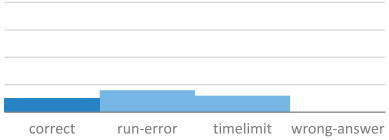


Black Out

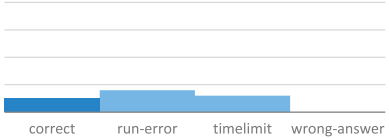
Greedy solution

- Use first move to black out one of the rows
- This makes the effective board dimensions even
- Now rotate every move of the opponent

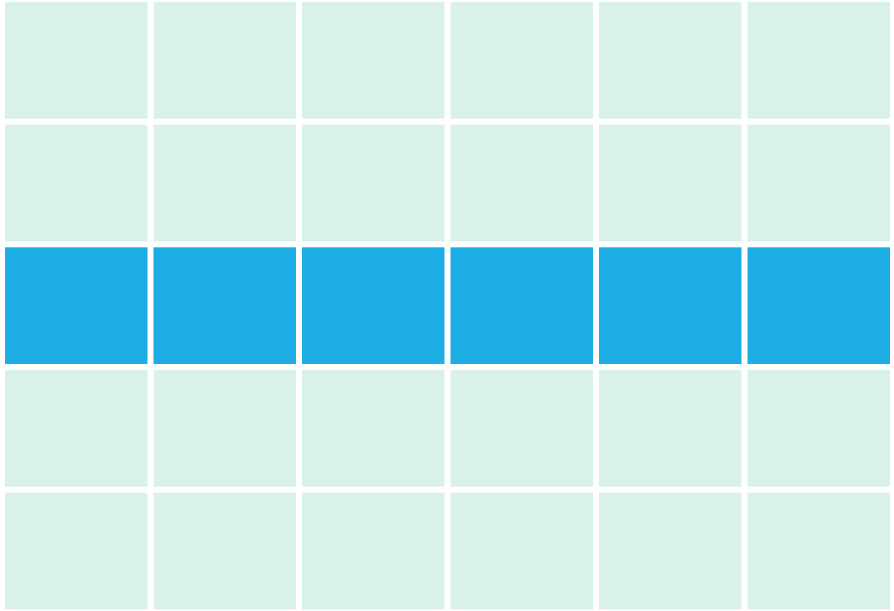


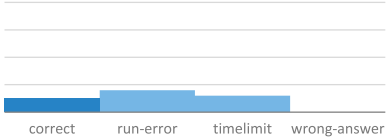


Black Out

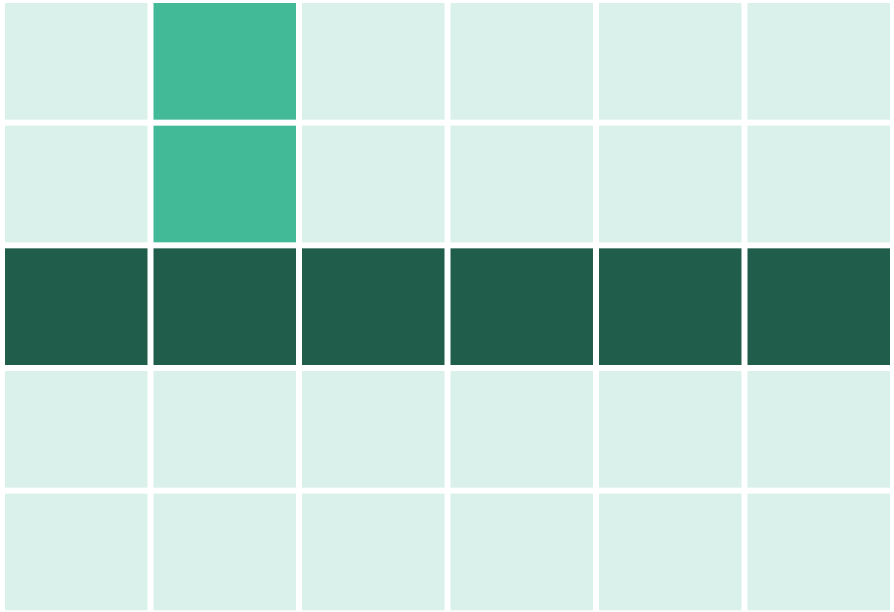


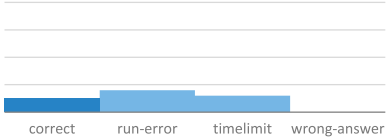
Black Out



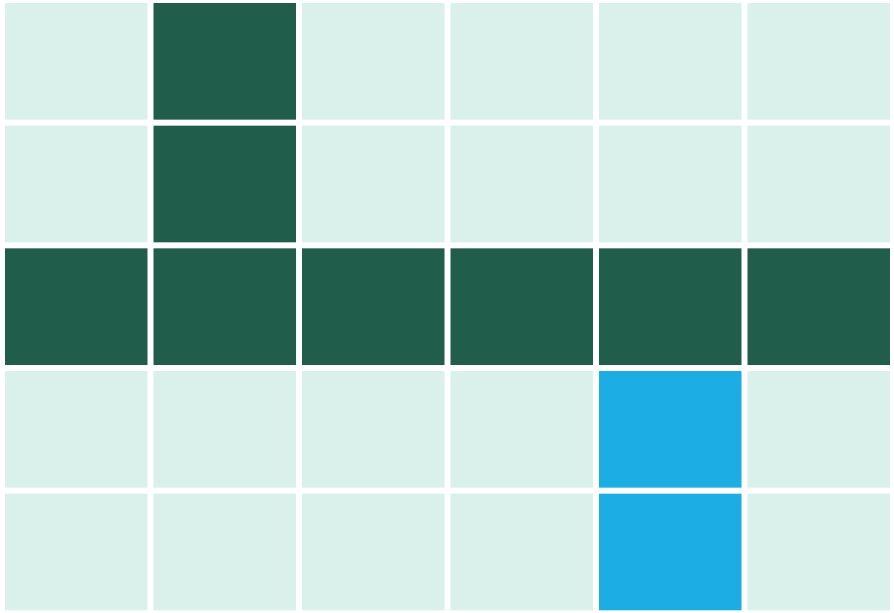


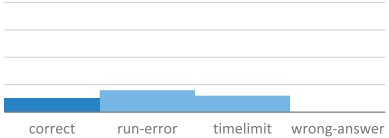
Black Out



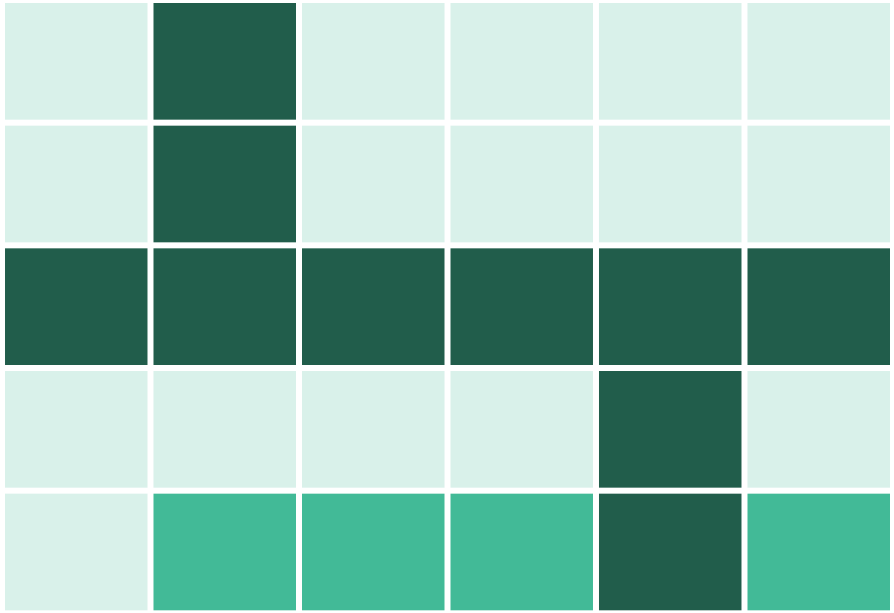


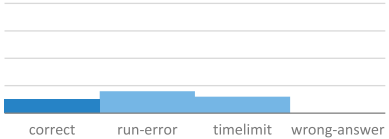
Black Out



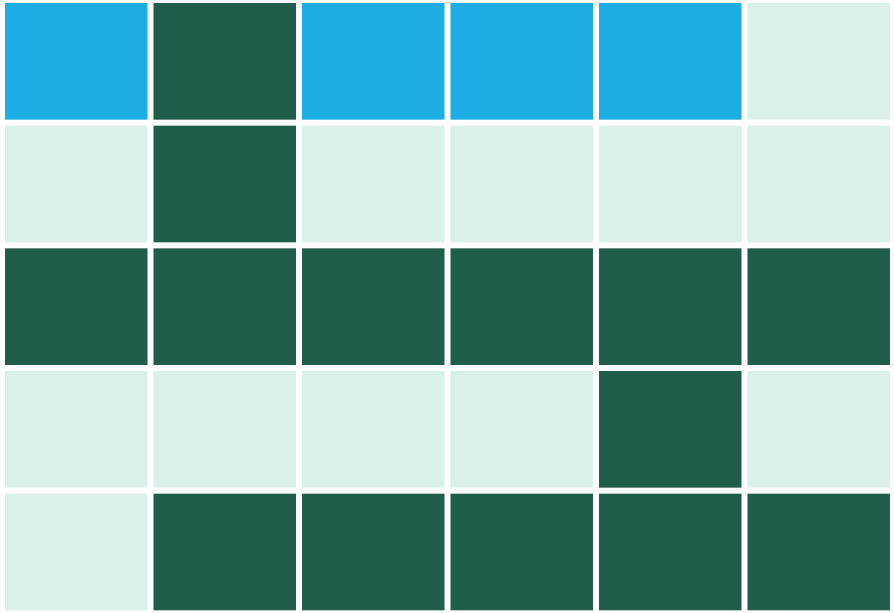


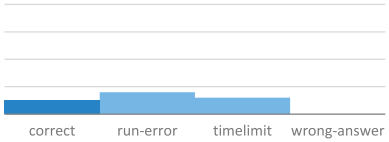
Black Out



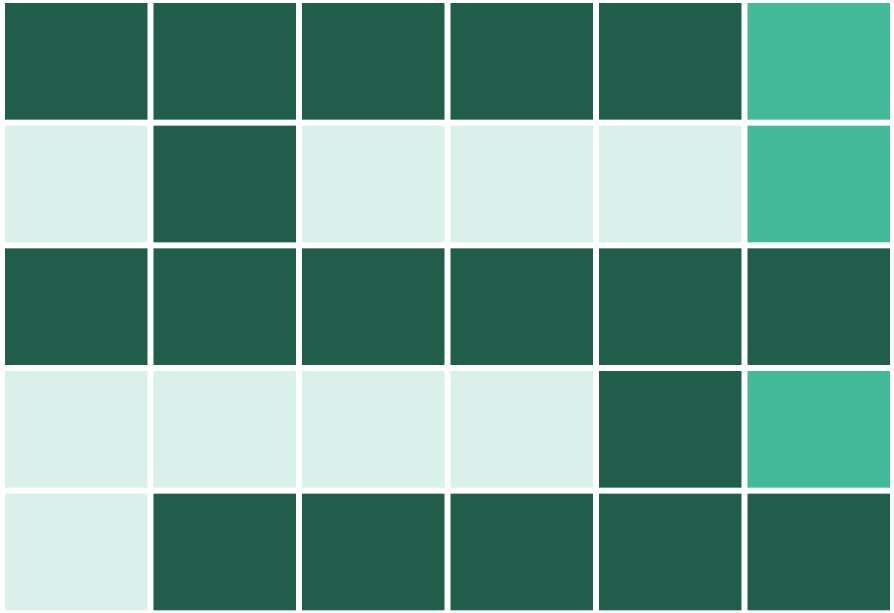


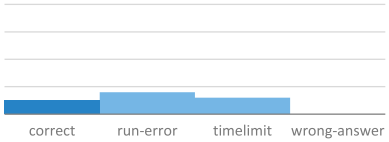
Black Out



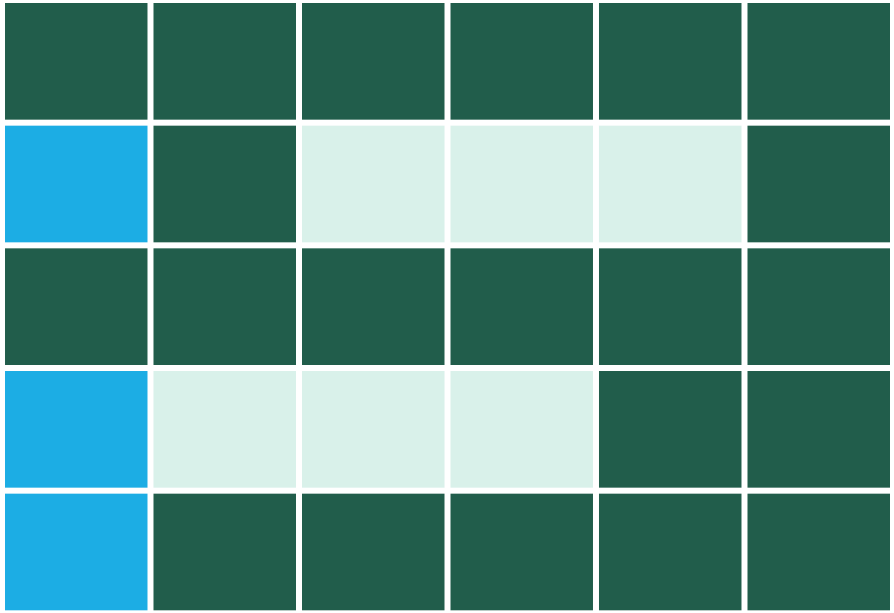


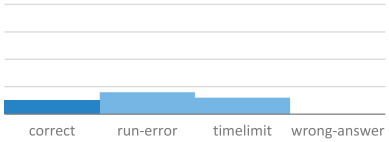
Black Out



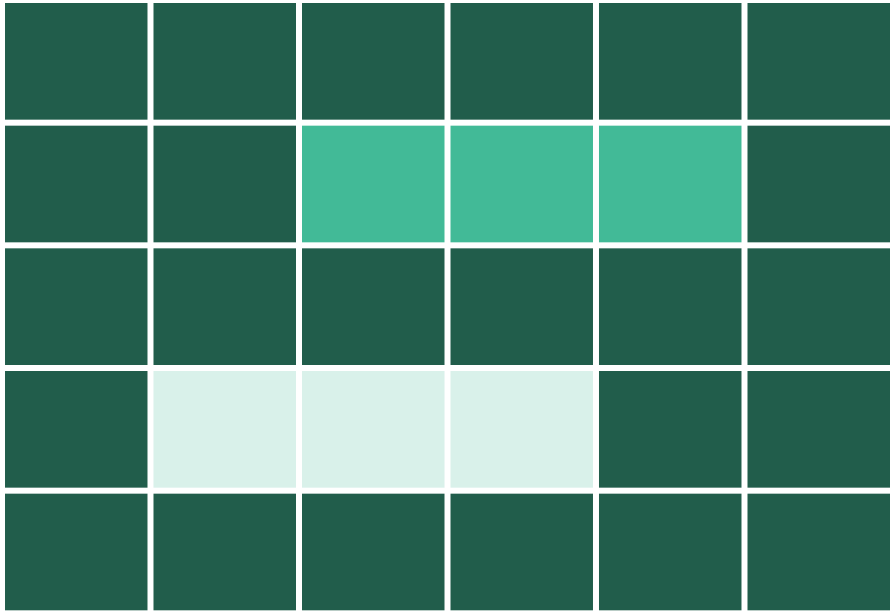


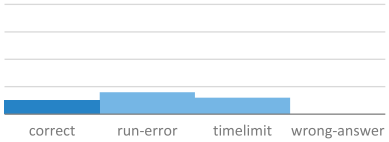
Black Out



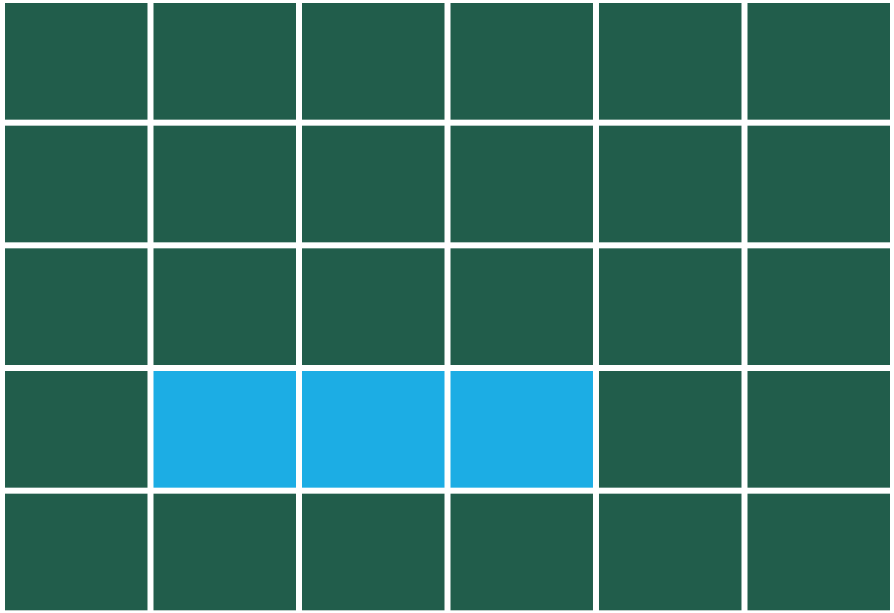


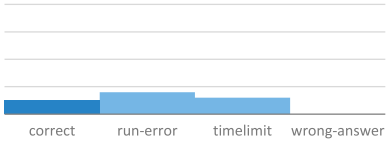
Black Out





Black Out





Black Out

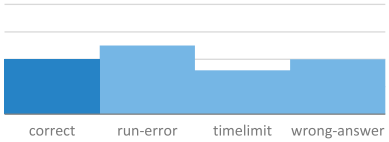
Greedy solution

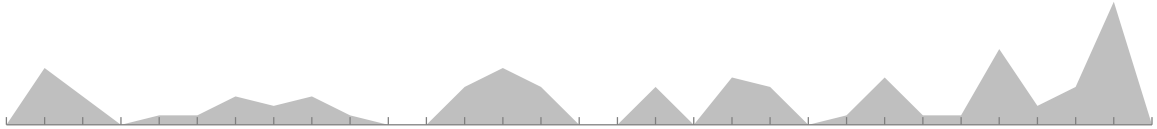
- Use first move to black out one of the rows
- This makes the effective board dimensions even
- Now rotate every move of the opponent

Also possible to **precompute** entire state space

- Much harder to program

$O(1)$ per move





Digit Sum

$O(b-a)$ is too slow!

Simplify: instead of $ds(a,b)$ do $ds(0,b) - ds(0,a-1)$

Lower the digits to 0 one by one starting on the right

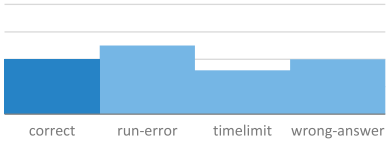
- For example to lower 25300 to 25000:

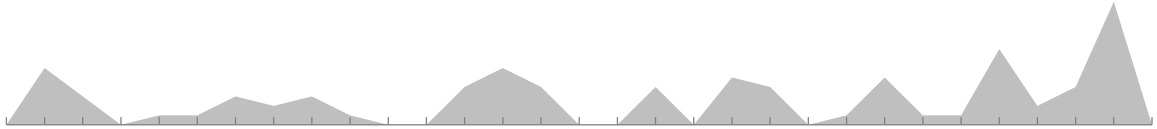
$$\underbrace{25}\underbrace{300}$$

$$(2+5)\cdot 300 \quad + \quad (2+1+0)\cdot 100 \quad + \quad 3\cdot 2\cdot 10\cdot(0+1+2+3+4+5+6+7+8+9)$$

Watch out for overflow

$O(\log b)$





Digit Sum

$O(b-a)$ is too slow!

Simplify: instead of $ds(a,b)$ do $ds(0,b) - ds(0,a-1)$

Lower the digits to 0 one by one starting on the right

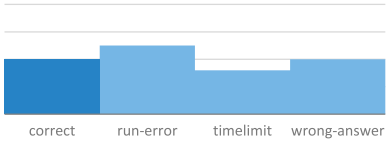
- For example to lower 25300 to 25000:

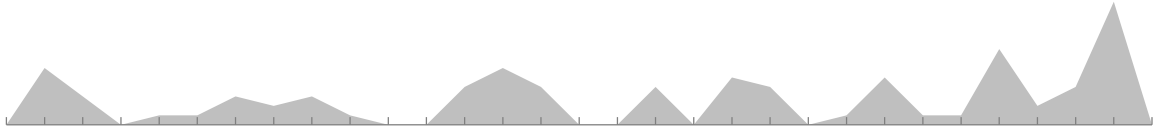
$$25300$$

$$(2+5)\cdot 300 \quad + \quad (2+1+0)\cdot 100 \quad + \quad 3\cdot 2\cdot 10\cdot(0+1+2+3+4+5+6+7+8+9)$$

Watch out for overflow

$O(\log b)$





Digit Sum

$O(b-a)$ is too slow!

Simplify: instead of $ds(a,b)$ do $ds(0,b) - ds(0,a-1)$

Lower the digits to 0 one by one starting on the right

- For example to lower 25300 to 25000:

$$25300$$

$$(2+5)\cdot 300 \quad + \quad (2+1+0)\cdot 100 \quad + \quad 3\cdot 2\cdot 10\cdot(0+1+2+3+4+5+6+7+8+9)$$

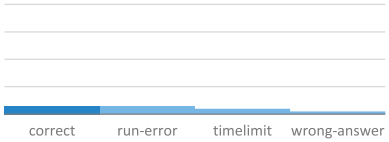
Watch out for overflow

$O(\log b)$

Another Dice Game

```
solve n = [undefined
    , 247073176590943186439168
    , 247073176590943186439168
    , 247073176590943186439168
    , 247073176590943186439168
    , 247073176590943186439168
    , 247072845992125906439168
    , 247070813693717458439168
    , 247066755953764920839168
    , 247057349205125525039168
    , 247033689015563322839168
    , 246972918870264891599168
    , 246861685382219625635168
    , 246649942550172329918528
    , 246186412211125491061952
    , 245228224202749711059488
    , 243985945560913777006648
    , 241881462878144723956828
    , 238716451221474026524964
    , 234321237560142152844748
    , 228531034766423460935995
    continued in next column
```

```
    , 222103036346998935592602
    , 212795095987221162977007
    , 200891157701282775146124
    , 186126554624712816982558
    , 169203799155206602426722
    , 150963413026932092081731
    , 130511956302057992471944
    , 108292622641792927940379
    , 86125966158468487939539
    , 65606605930495666414544
    , 48253097635645381932276
    , 33777682071351065695944
    , 21581704251132856036848
    , 13086424949502634295472
    , 7189532461169824483008
    , 3899282638457082901248
    , 1865911064934695745024
    , 667911312560821570560
    , 237729048123966382080
    , 36338228665926746112
    ]!!n /
24870815968582502763724 8
```



Another Dice Game

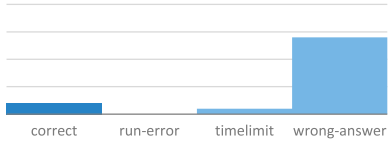
Either **simulate** all possible games with some tricks

- State: number of points, number of dice, subset of values left
- Use memoization to avoid redundant work
- With 8 dice only 1287 possible rolls (not enough time for $6^8 \approx 1.7\text{M}$)

$O(40 \cdot 8 \cdot 2^6 \cdot 1287)$

Or **precompute** team-side :)

- Only 40 inputs possible
- Even without smart dice rolls this takes less than 10 seconds

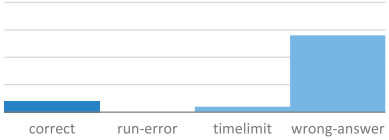


Chess Competition

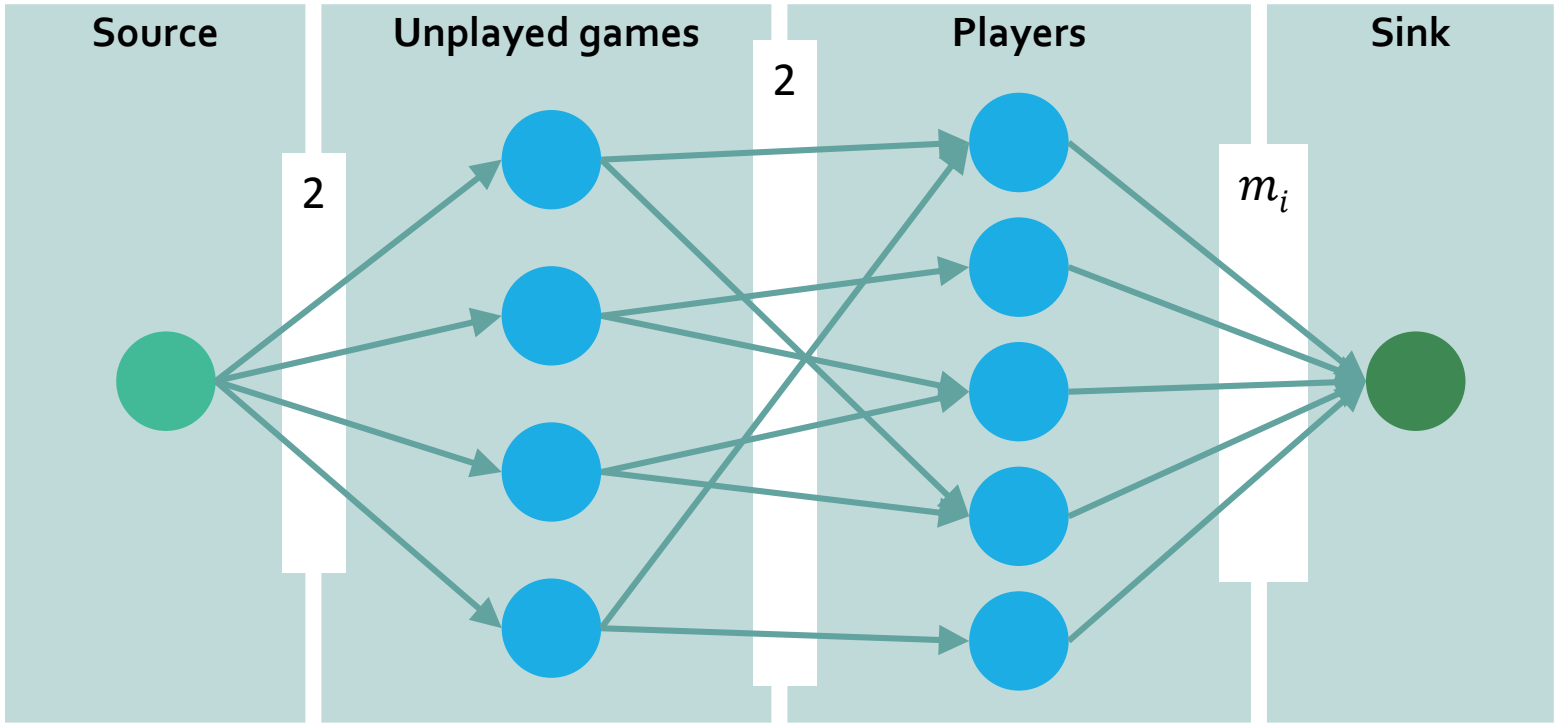
For each player try to let him/her win

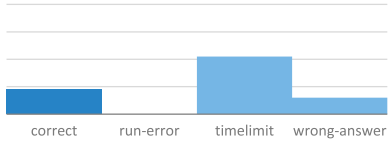
- Assume he/she wins all his/her remaining matches
- The points of the unplayed games should be distributed among other players
- But: in such way that nobody scores more points than this player
- Use Maximum Network Flow algorithm

$O(n^5)$



Chess Competition





John's Book Stack

Pulling out a book in the sorted top is never useful, so always pull out the first non-sorted book

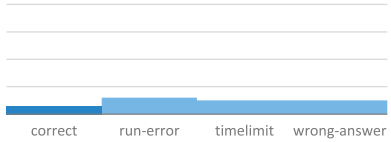
However, simulation is too slow!

To move a big book from the top to position i, takes $2^i - 1$ steps

- (when there are no duplicates)
- Be a bit clever with duplicates

Watch out for overflow

$O(n)$



Hot Dogs in Manhattan

BFS to find minimum distance for each gridpoint

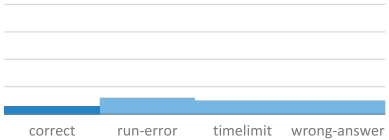
Binary search over answer x

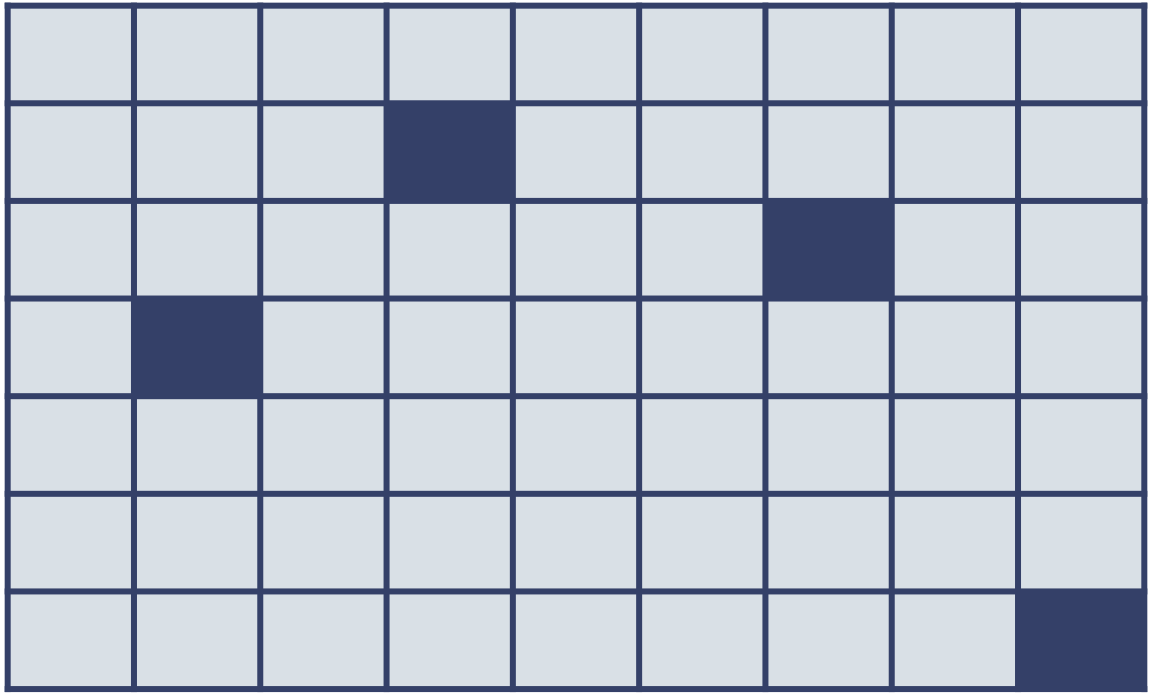
Given an optimal solution, the leftmost point can be moved further to the left without loss of optimality if it doesn't move closer than x to another stand

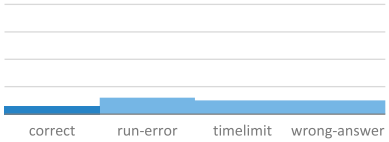
- Same for the rightmost point to the right
- So: we only need to consider the leftmost and rightmost point with $\geq x$ per row

Now for each pair of rows, check for combination of leftmost and rightmost point

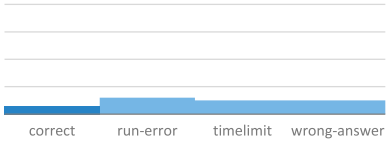
$O(w \cdot h \cdot \log(w \cdot h))$


correct
run-error
timelimit
wrong-answer

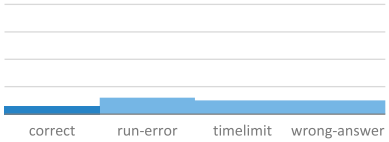




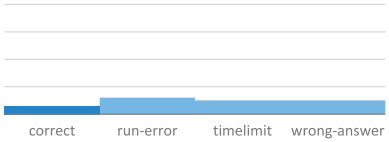
4	3	2	1	2	3	2	3	4
3	2	1		1	2	1	2	3
2	1	2	1	2	1		1	2
1		1	2	3	2	1	2	3
2	1	2	3	4	3	2	3	2
3	2	3	4	5	4	3	2	1
4	3	4	5	4	3	2	1	



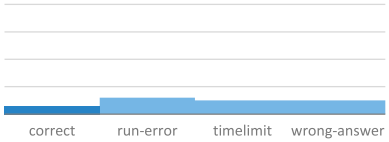
4	3	2	1	2	3	2	3	4
3	2	1		1	2	1	2	3
2	1	2	1	2	1		1	2
1		1	2	3	2	1	2	3
2	1	2	3	4	3	2	3	2
3	2	3	4	5	4	3	2	1
4	3	4	5	4	3	2	1	



4	3	2	1	2	3	2	3	4
3	2	1		1	2	1	2	3
2	1	2	1	2	1		1	2
1		1	2	3	2	1	2	3
2	1	2	3	4	3	2	3	2
3	2	3	4	5	4	3	2	1
4	3	4	5	4	3	2	1	

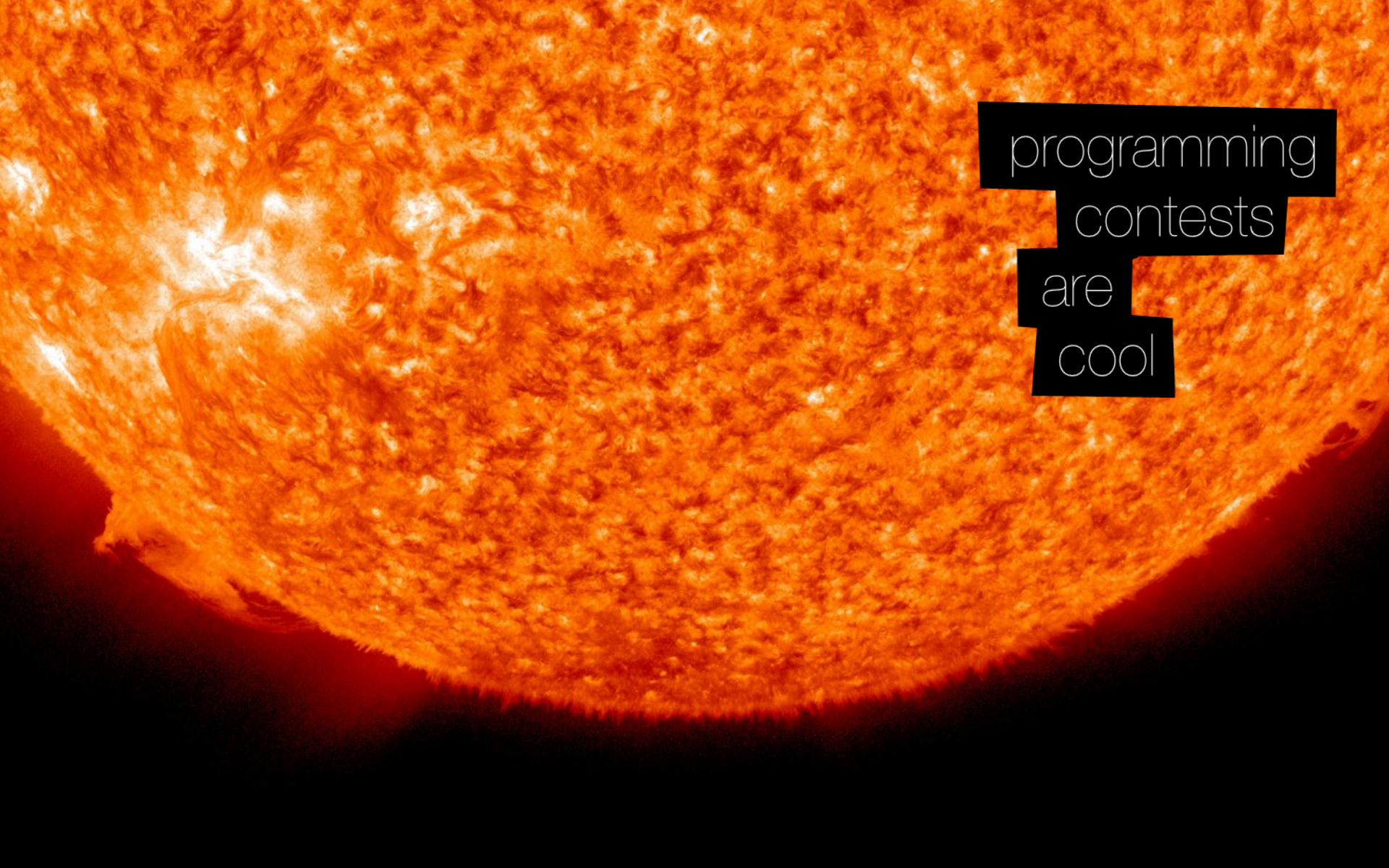


4	3	2	1	2	3	2	3	4
3	2	1		1	2	1	2	3
2	1	2	1	2	1		1	2
1		1	2	3	2	1	2	3
2	1	2	3	4	3	2	3	2
3	2	3	4	5	4	3	2	1
4	3	4	5	4	3	2	1	



4	3	2	1	2	3	2	3	4
3	2	1		1	2	1	2	3
2	1	2	1	2	1		1	2
1		1	2	3	2	1	2	3
2	1	2	3	4	3	2	3	2
3	2	3	4	5	4	3	2	1
4	3	4	5	4	3	2	1	

Scores


programming
contests
are
cool

#	AFFIL.	TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
1		team5	9	1424	1 (204 + 0)	2	2 (122 + 20)	1 (14 + 0)	1 (208 + 0)	1 (140 + 0)	1 (208 + 0)	1 (168 + 0)	2 (210 + 20)	2 (90 + 20)
2		King High	8	980	0	2 (256 + 20)	1 (121 + 0)	1 (25 + 0)	1 (34 + 0)	2 (88 + 20)	3 (67 + 40)	3 (203 + 40)	2 (46 + 20)	1
3		ASDF	7	763	0	2 (174 + 20)	0	1 (137 + 0)	1 (12 + 0)	2 (52 + 20)	1 (215 + 0)	0	1 (22 + 0)	1 (111 + 0)
4		Geen Syntax	7	927	0	1 (164 + 0)	2	2 (132 + 20)	1 (15 + 0)	6 (106 + 100)	1 (36 + 0)	2	2 (37 + 20)	2 (277 + 20)
5		Sudo Win	6	649	0	0	3	1 (72 + 0)	1 (38 + 0)	3 (87 + 40)	2 (113 + 20)	0	1 (47 + 0)	2 (212 + 20)
6		Classy Carl & the Nullpointers	6	934	1	1	3	3 (148 + 40)	1 (34 + 0)	2 (93 + 20)	2 (294 + 20)	1	2 (84 + 20)	1 (181 + 0)
7		Algorithmics Anonymous	5	557	1	0	0	4 (235 + 60)	1 (16 + 0)	2 (82 + 20)	1 (61 + 0)	0	3 (43 + 40)	1
8	TU/e	Vomit	4	352	1	0	0	1 (133 + 0)	1 (14 + 0)	4	1 (164 + 0)	0	1 (41 + 0)	1
9		The Redeem Team	4	423	1	1	0	1	1 (34 + 0)	6	2 (45 + 20)	0	3 (101 + 40)	1 (183 + 0)
10	TU/e	metCake!	4	831	0	2 (155 + 20)	0	6 (282 + 100)	2 (27 + 20)	0	7	0	5 (147 + 80)	0
11	TU/e	U+FFFD	3	219	0	0	0	2	1 (22 + 0)	5	3 (104 + 40)	0	2 (33 + 20)	0
12		Team Amersfoort	3	232	0	0	0	3	1 (61 + 0)	1 (97 + 0)	0	0	2 (54 + 20)	2
13		Team Integer	3	247	0	0	0	1	1 (18 + 0)	2	2 (155 + 20)	0	1 (54 + 0)	0
14		Infinite Improbability Drive	3	295	0	2	0	0	1 (15 + 0)	1 (118 + 0)	1	0	5 (82 + 80)	0
15		Macbookpro	3	359	0	0	0	2 (190 + 20)	1 (21 + 0)	0	0	0	4 (68 + 60)	0
16		Kale koppen niet te stoppen	3	371	0	0	0	3 (209 + 40)	1 (17 + 0)	2	0	0	3 (65 + 40)	0
17		iapc	3	379	0	0	0	2 (207 + 20)	1 (16 + 0)	2	0	0	4 (76 + 60)	1
18		Mutable Void	3	450	0	0	0	0	1 (40 + 0)	0	5 (187 + 80)	0	2 (123 + 20)	0
19		Ololiuhqui	3	803	0	0	0	2 (149 + 20)	1 (21 + 0)	0	0	0	18 (273 + 340)	0
20		Team Spirit	2	141	1	1	4	1	1 (48 + 0)	1	6	1	4 (33 + 60)	1
21		Team Fietspomp	2	148	0	2	4	4	2 (28 + 20)	0	1	0	3 (60 + 40)	0
22		3g0 > br4in5	2	162	0	0	3	0	1 (69 + 0)	3	0	0	2 (73 + 20)	0
23		insertnamehere	2	164	0	0	0	0	1 (62 + 0)	0	0	0	1 (102 + 0)	1
24		veni vidi velcro	2	201	0	0	0	2	1 (26 + 0)	1	0	0	3 (135 + 40)	0
25		Concurrent Read Write	2	205	0	0	0	5 (85 + 80)	1 (40 + 0)	0	3	0	12	0
26		davini	2	263	0	0	0	2	1 (59 + 0)	3	0	0	4 (144 + 60)	0
27		The Nullpointers	2	271	0	0	0	1	2 (35 + 20)	0	0	0	5 (136 + 80)	5
28		Het Pino Trio	2	452	0	0	0	0	1 (67 + 0)	0	0	0	8 (245 + 140)	0
29		Team TnT	1	184	0	0	0	0	1 (184 + 0)	3	0	0	7	0
30		C Hekje	1	299	0	0	0	3	8 (159 + 140)	0	0	0	0	0
31	TU/e	phineas and ferb	0	0	0	0	0	1	8	0	0	0	2	1

WHEN YOU COULD'VE GONE HOME