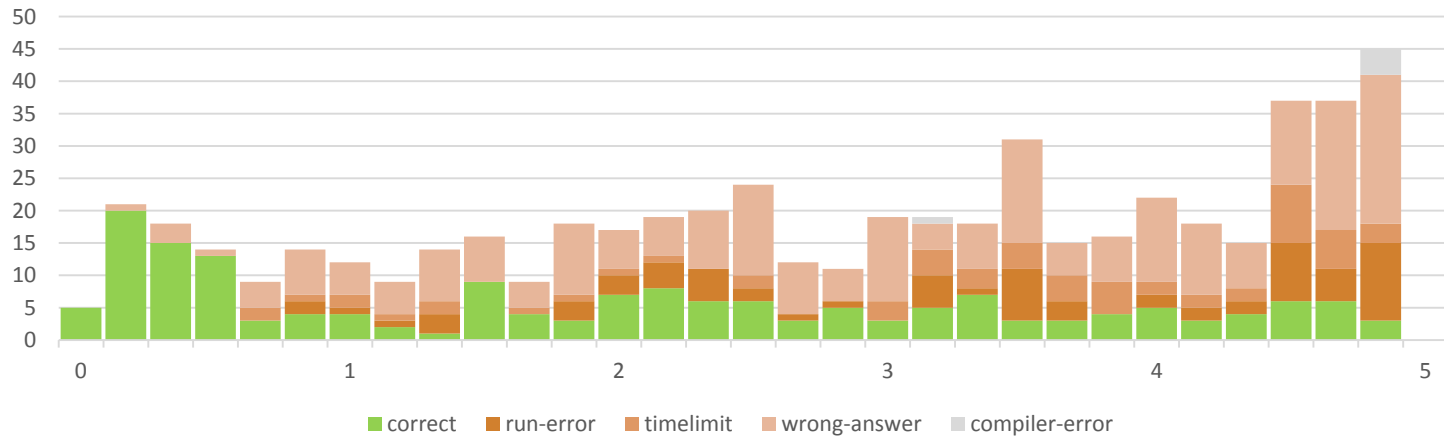


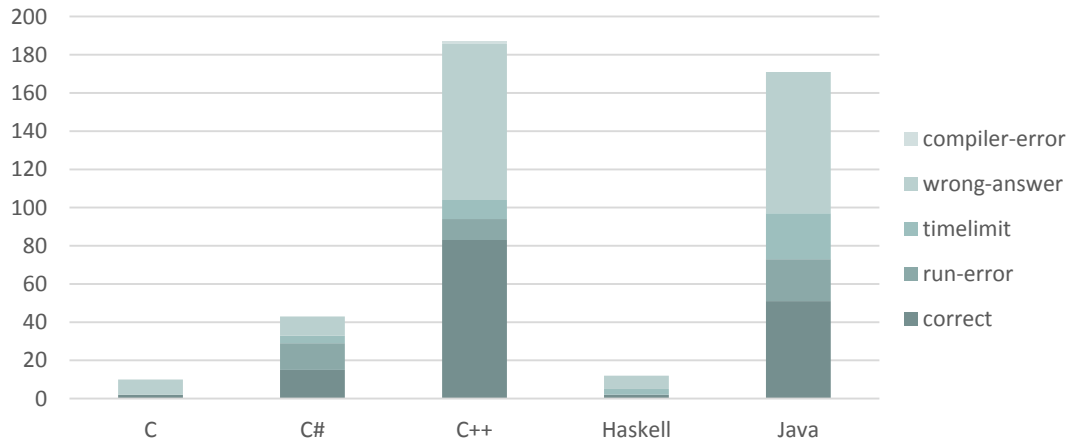
BAPC 2013

STATS + SOLUTIONS + SCORES

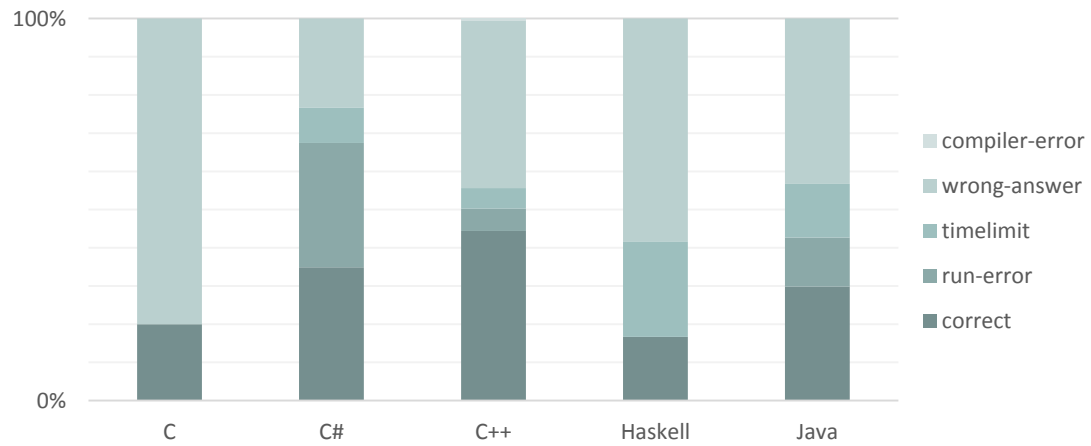
SUBMISSIONS OVER TIME



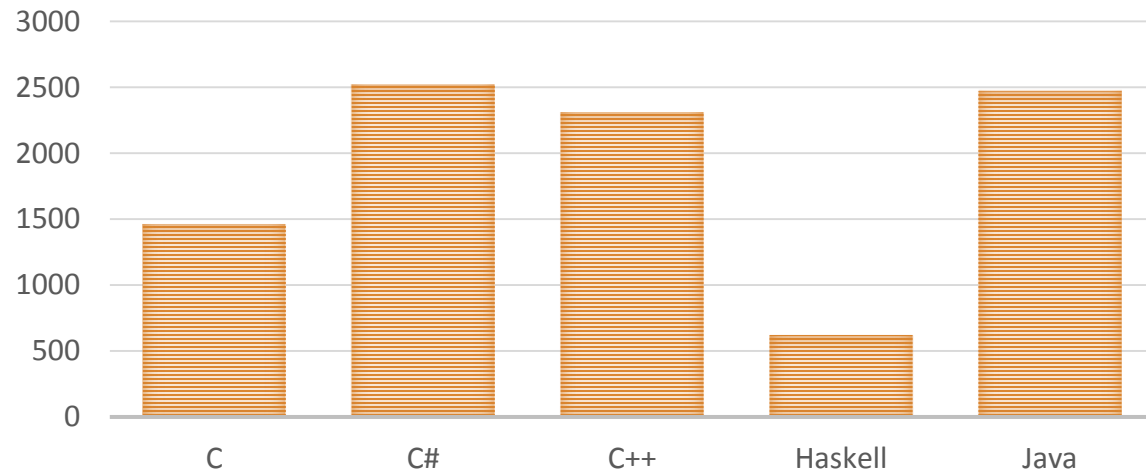
SUBMISSIONS BY LANGUAGE



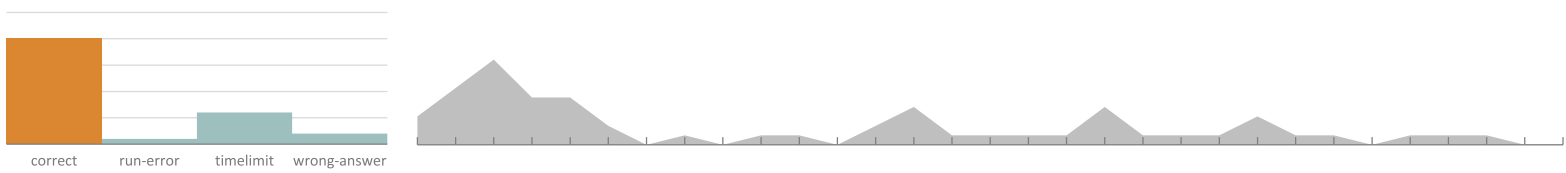
SUBMISSIONS BY LANGUAGE



AVERAGE CODE LENGTH



Solutions



Flying Safely

Minimum spanning tree with weight 1 for each edge

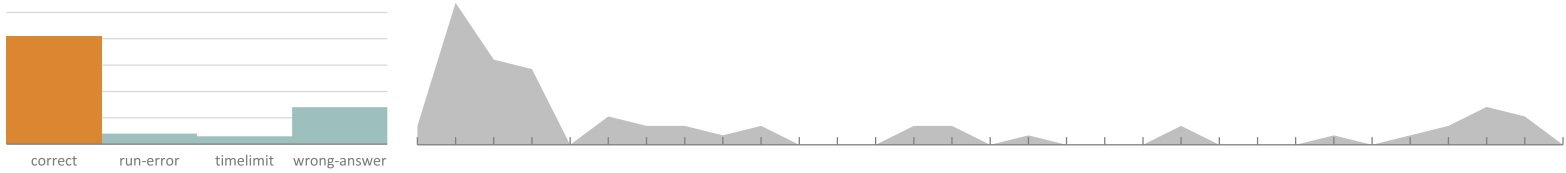
Tree of n nodes has $n - 1$ edges

Forget the spanning tree and output $n - 1$

$O(1)$

- + $O(m)$ to read input





Incognito

Group attributes per category

- **Hashmap**

Possibilities for each category

- No item
- Exactly 1 item

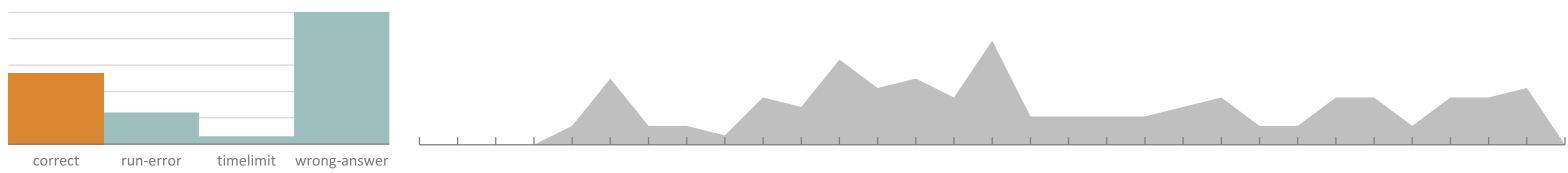
For n items: $n + 1$ possibilities

Use **multiplication**

- Subtract 1 for the no disguise

$O(n)$





Administrative Difficulties

Just **do** the bookkeeping

- Maybe using a **hashmap**

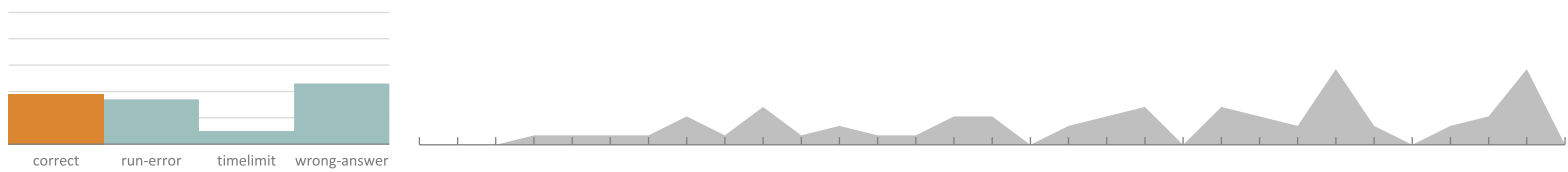
Don't do it wrong

- $[42000000, 01] = 42000001$

$O(n + m)$



– team "Royals"



Destination Unknown

Run **Dijkstra's** from s , f and g

Check 'triangle equality' for each destination i

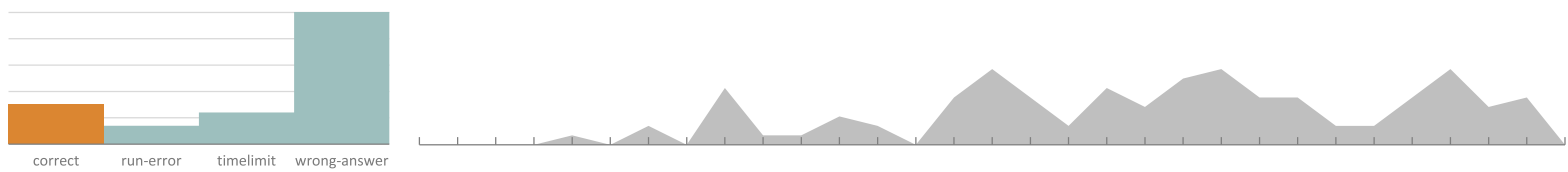
- $d(s, f) + d(f, g) + d(g, i) = d(s, i)$ or
- $d(s, g) + d(g, f) + d(f, i) = d(s, i)$

More ideas

- Keep extra state "seen (f, g) " in your Dijkstra's
- Double all weights (carefully), subtract 1 from (f, g) , is distance odd?
- Use Dijkstra's from s to make a shortest-path-DAG and do a BFS from (f, g)

$O(n^2)$





Cracking the Code

Check for each encrypted message if it could match

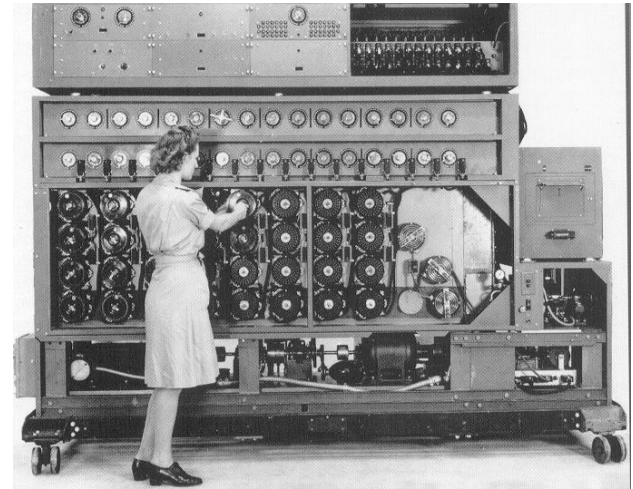
- Walk through the string
 - If you've seen a letter before, it should map to the same letter
- Check both decoded to encrypted and encrypted to decoded
- If it matches, save the matching of the letters

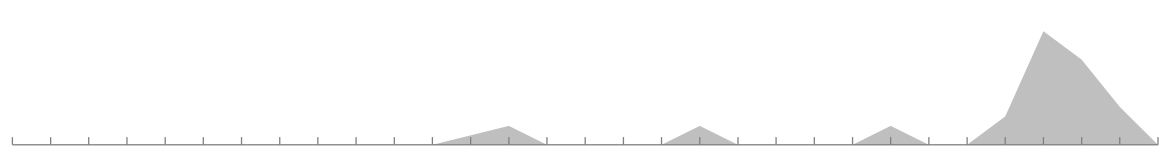
Print "?" when there is not exactly 1 matching for this letter

Don't forget

- When 25 letters are known, so is the 26th!

O(size of input)





Jailbreak

There must be a 'splitting' point

- Where paths converge from outside, prisoner 1 and prisoner 2

Three **BFS**'s

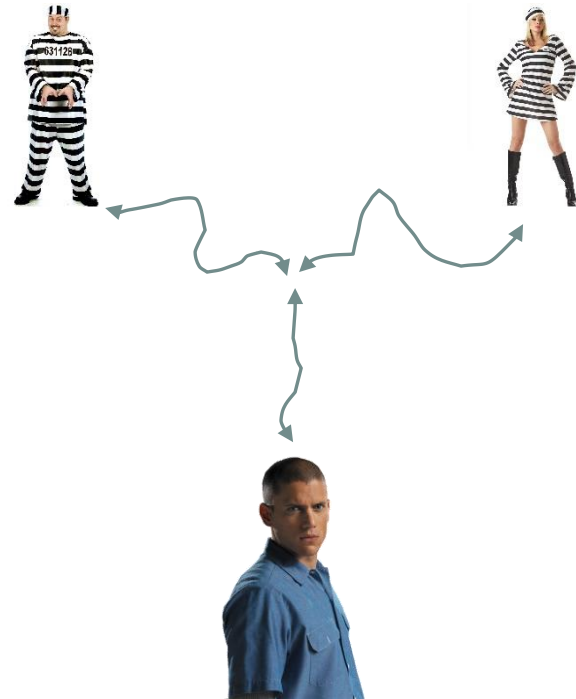
- From outside, prisoner 1 and prisoner 2

Splitting point has minimum sum of these three distances

Careful

- Splitting point might be a door

$O(h * w)$





Bribe

Memoization / dynamic-programming

Calculate probability of success, for each

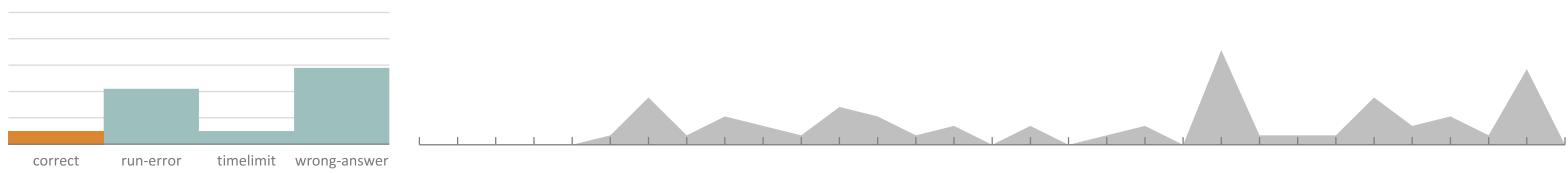
- Amount of henchmen already converted
- Subset of henchmen already asked
 - Could use a bitmask

Money spent can be deduced from that subset

Answer is probability with 0 converted, \emptyset already asked

$$O(2^n * n^2)$$





Getting Through

Binary search over radius

Check if radius r is possible

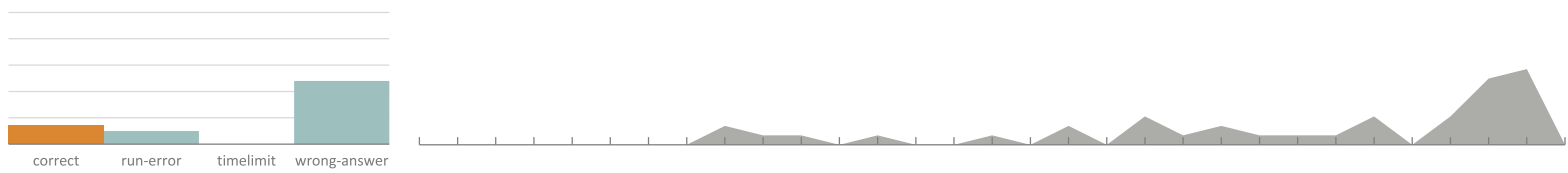
- Extend all circles with r
- Move walls by r
- Possible iff no path from left to right wall via overlapping circles

$$O(\log 10^{5-6} * n^2)$$

Or use a **minimum spanning tree**

$$O(n^2)$$





Hidden Camera

Trick

- Move camera to origin
- Rotate such that base wall is on x-axis
- Construct boundary lines and find intersection points

Calculate both areas

- Look in your cheat sheet...

Careful with floating points, small rounding errors can lead to finding no intersection point at all



Encoded Coordinates

Introduce new name for $H(n - 1)$

- $I(n + 1) = H(n)$

Looks like a matrix multiplication right?

- $$\begin{pmatrix} F(n + 1) \\ G(n + 1) \\ H(n + 1) \\ I(n + 1) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ K & 0 & 0 & 1 \\ 1 & K & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} F(n) \\ G(n) \\ H(n) \\ I(n) \end{pmatrix}$$

$$\mathbf{v}_{n+1} = A\mathbf{v}_n \Rightarrow \mathbf{v}_N = A^{N-1}\mathbf{v}_1$$

Find $H(0)$ with x and use **matrix exponentiation** to find N^{th} element































$O(\log N + P)$

Can you do it in $O(\log N + \log P)$?

Scores

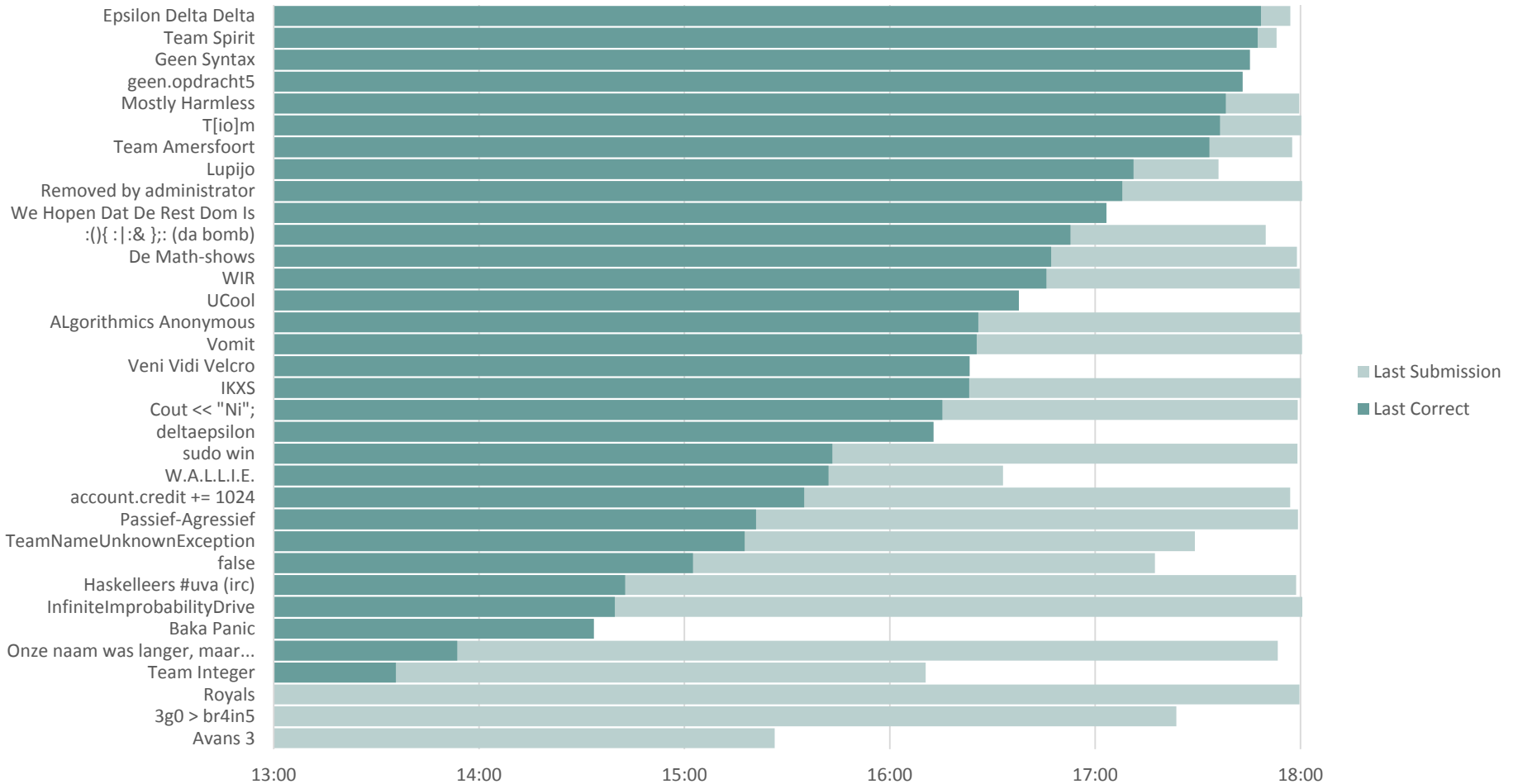
Scoreboard BAPC 2013

final standings

#	AFFIL.	TEAM	SCORE	A ○	B ●	C ●	D ○	E ●	F ●	G ●	H ●	I ●	J ●	
▼1	 	Geen Syntax	10	1331	2 (80 + 20)	1 (130 + 0)	2 (69 + 20)	1 (95 + 0)	1 (172 + 0)	1 (12 + 0)	4 (244 + 60)	3 (226 + 40)	1 (7 + 0)	2 (136 + 20)
▼2	 	geen.opdracht5	10	1654	3 (116 + 40)	1 (206 + 0)	3 (195 + 40)	1 (31 + 0)	1 (238 + 0)	1 (18 + 0)	2 (257 + 20)	4 (283 + 60)	1 (16 + 0)	1 (134 + 0)
▼3	 	ALgorithmics Anonymous	6	635	1 (42 + 0)	1 (205 + 0)	1 (143 + 0)	3 (187 + 40)	0	1 (6 + 0)	19	0	1 (12 + 0)	4
▼4	 	UCool	6	845	1 (93 + 0)	0	3 (209 + 40)	2 (217 + 20)	1 (196 + 0)	1 (44 + 0)	0	0	1 (26 + 0)	0
▼5	 	Mostly Harmless	6	976	1 (113 + 0)	2	2 (151 + 20)	7 (230 + 120)	0	1 (15 + 0)	0	0	1 (9 + 0)	3 (278 + 40)
▼6	 	sudo win	5	574	5 (152 + 80)	0	1	1 (49 + 0)	0	1 (19 + 0)	6 (163 + 100)	5	1 (11 + 0)	0
▼7	 	Epsilon Delta Delta	5	592	1 (133 + 0)	0	4	1 (90 + 0)	3 (288 + 40)	1 (23 + 0)	4	0	1 (18 + 0)	0
▼8	 	We Hopen Dat De Rest Dom Is	5	685	1 (127 + 0)	0	4 (213 + 60)	1 (243 + 0)	0	1 (30 + 0)	0	0	1 (12 + 0)	0
▼9	 	Cout << "Ni";	4	421	2 (159 + 20)	1	2	1 (195 + 0)	0	1 (28 + 0)	0	0	1 (19 + 0)	0
▼10	 	Vomit	4	459	5 (205 + 80)	1	4	1 (154 + 0)	0	1 (8 + 0)	0	2	1 (12 + 0)	0
▼11	 	WIR	4	462	3 (225 + 40)	4	3	5	0	1 (30 + 0)	0	2 (124 + 20)	1 (23 + 0)	0
▼12	 	Team Amersfoort	4	490	1 (120 + 0)	0	3 (273 + 40)	7	0	1 (26 + 0)	0	0	1 (31 + 0)	0
▼13	 	Veni Vidi Velcro	4	675	1 (118 + 0)	0	3 (203 + 40)	0	0	1 (184 + 0)	0	0	1 (130 + 0)	0
▼14	 	Onze naam was langer, maar...	3	86	1 (53 + 0)	1	1	2	0	1 (10 + 0)	0	3	1 (23 + 0)	0
▼15	 	Removed by administrator	3	294	5	0	2	1 (247 + 0)	0	1 (20 + 0)	0	0	1 (27 + 0)	0

...

WHEN YOU COULD'VE GONE HOME...



We had fun :)

We hope you did too.

Let us know what you think.

See you next year!
And we hope you agree...



programming

contests

are

cool