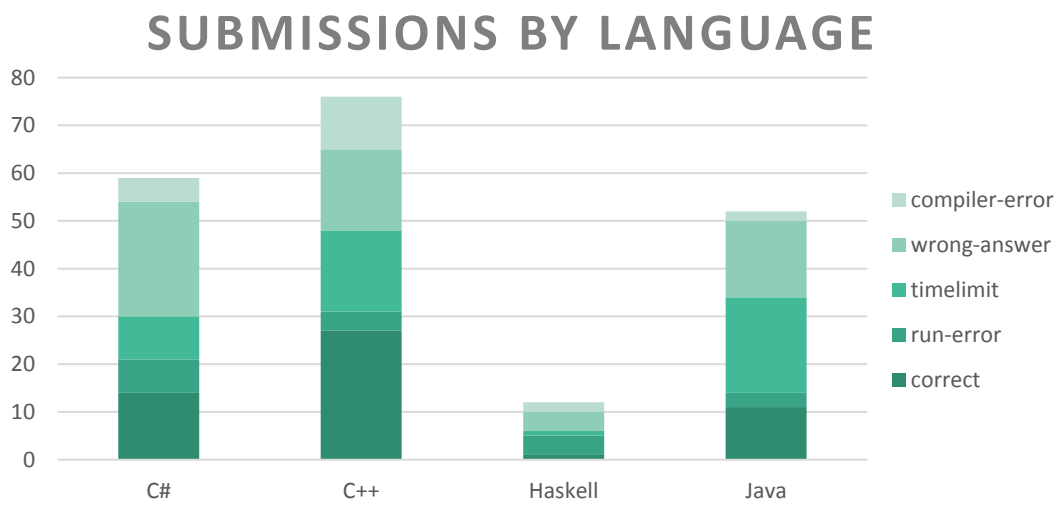
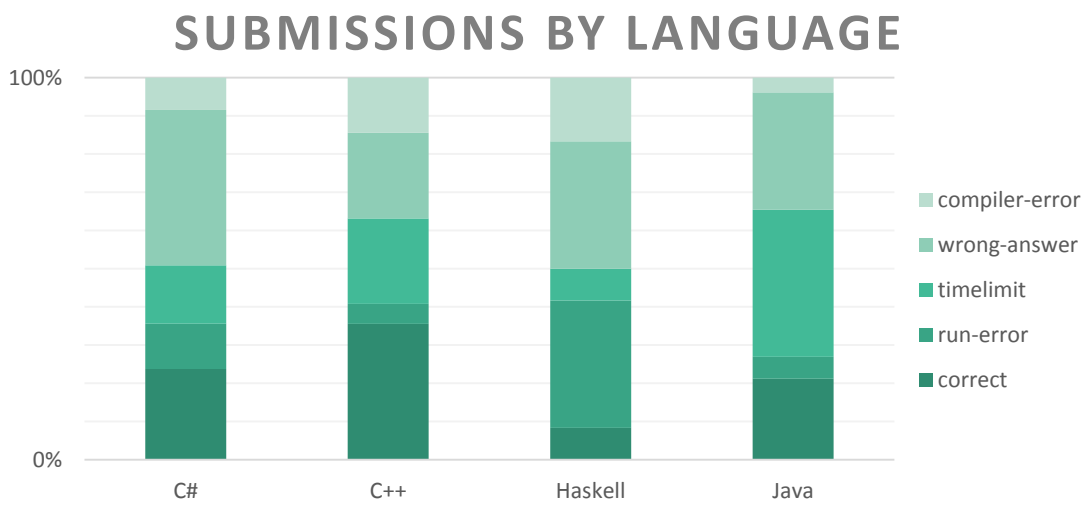
BAPC Preliminaries 2012

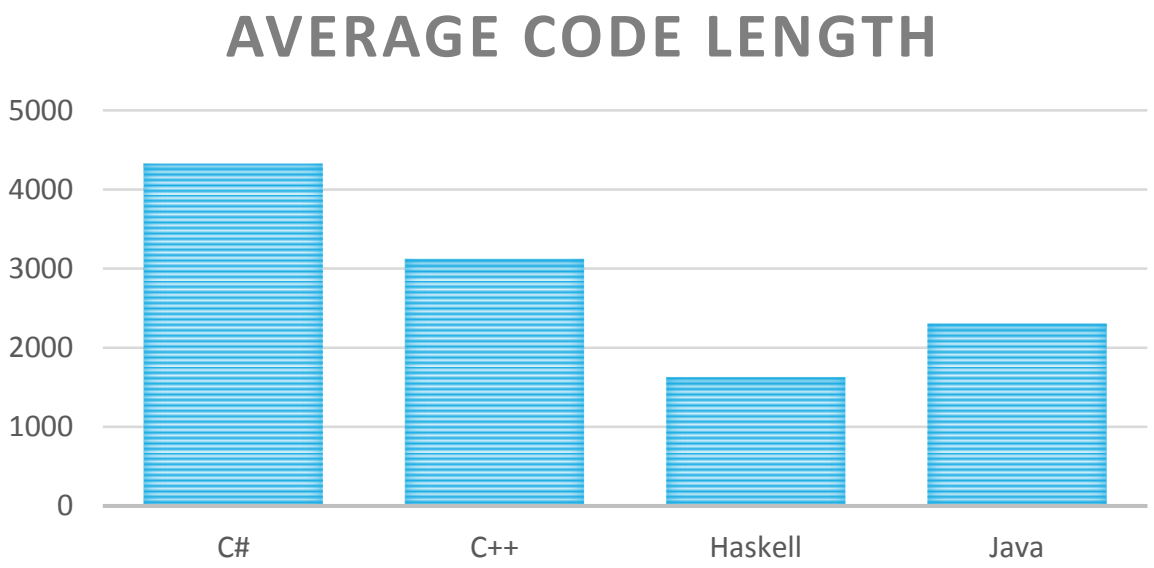
STATS + SOLUTIONS + SCORES

SUBMISSIONS OVER TIME

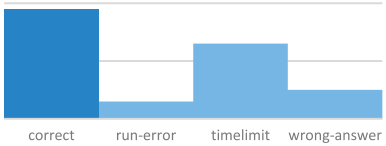
** STATS FROM UTRECHT + TWENTE*


SUBMISSIONS BY LANGUAGE
80
70
60
50
40
30
20
10
0
C#
C++
Haskell
Java
compiler-error
wrong-answer
timelimit
run-error
correct


SUBMISSIONS BY LANGUAGE
100%
0%
C#
C++
Haskell
Java
compiler-error
wrong-answer
timelimit
run-error
correct


AVERAGE CODE LENGTH
5000
4000
3000
2000
1000
0
C#
C++
Haskell
Java

Solutions



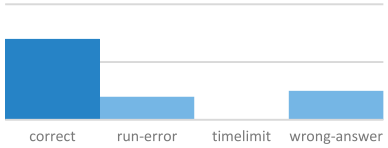
Annoying Mosquitos

Don't be tempted to loop through all gridpoints in a swat

For each swat, check which mosquitos are in range

- Don’t bother with quadtrees, this is fast enough

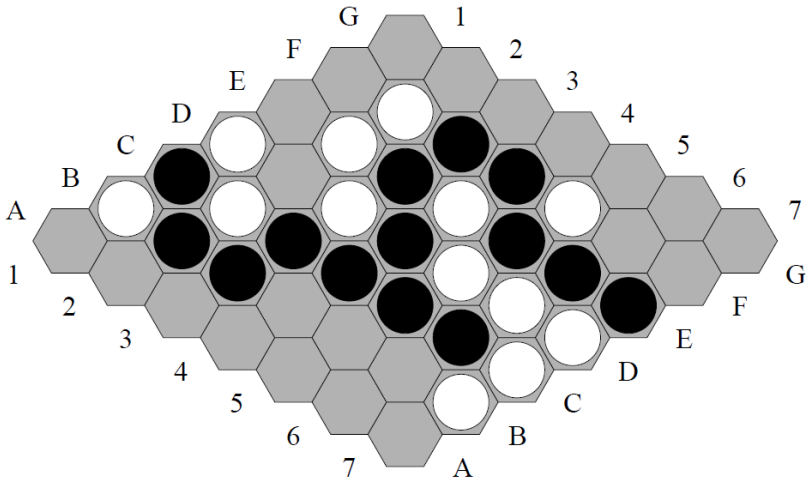
$O(nm)$

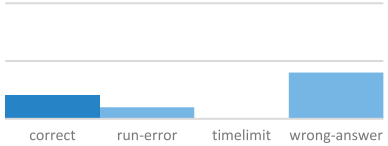


Hex

Twice a **Depth First Search** or **Breadth First Search**

$O(n^2)$





Cubing

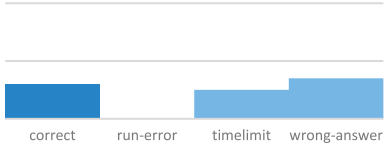
Just **simulate**

- But tricky to get all cases correct!

Possible strategies

- Represent the moves by a combination of a few (3) basic moves
- Don't explicitly program all the changing variables but represent them in an easily readable format (like string) and interpret that
- Geometric approach where every sticker is a point in 3d space

$O(n)$



Jumbled Letters

Normalize words for quick lookup

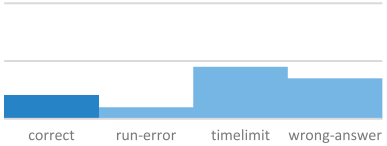
- For example by sorting the letters

Store all words in map (like **HashMap**) with normalized word as key

For each subset of the letters check if normalized version is present in map

Output the longest/alphabetically first

$O(2^k c)$ lookups (where k = the number of letters)



Float to Fraction

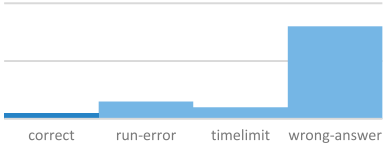
$$0.abcd(ef) = 0.abcd + 0.0000(ef)$$
$$= \frac{abcd}{10000} + \frac{0.(ef)}{10000}$$

$$0.(ef) \times 100 - 0.(ef) = ef$$
$$0.(ef) \times 99 = ef$$
$$0.(ef) = ef/99$$

$$= \frac{abcd}{10000} + \frac{ef/99}{10000}$$

Normalize above with **GCD** (Euclid's algorithm)

Use long's and **beware of overflow**!



Bad Scientist

Graph

- Node for each theory, edge for each contradiction

You may recognize the Maximum Independent Set problem

- State of the art is close to $O^*(1.3^n)$ but probably infeasible for contest

Dual to **Minimum Vertex Cover** which is **fixed parameter tractable**: $O^*(2^k)$

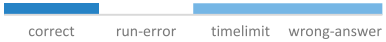
- So feasible since k is low

Multiple **branching** strategies possible, for example:

- Recursively visit all edges
  - If one of the nodes is already excluded, continue with next edge
  - Otherwise: branch
- Stop after all edges have been visited or recursion depth exceeds k

$O(2^k m)$

- Can even be improved to $O^*(1.3^k)$ but for this contest that was not necessary



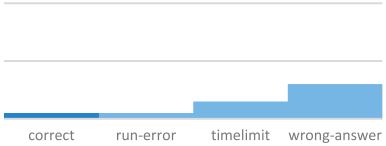
Islands

First, find minimum distance between each pair of islands

- Shortest distance is always between a vertex and an edge
- For each pair of vertex and edge, calculate line-to-point distance

Use your favorite **shortest path** algorithm to find the final route

$O(n^2 m^2)$



Extreme Shopping

Sort shops on price per item

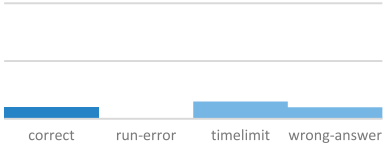
Dynamic programming

- State: (shop index, number of items to buy)

At each shop, if you decide to buy items there, it is best to buy as many as you can

$$f(i,j) = \text{best price considering only shop 1 through } i, \text{ buying } j \text{ items}$$
$$f(i,0) = 0$$
$$f(0,j) = \infty$$
$$f(i,j) = \min\big(f(i-1,j), b \cdot p_i + o_i + f(i-1, j-b)\big)$$
$$where\ b = \min(s_i, j)$$

$O(nm)$



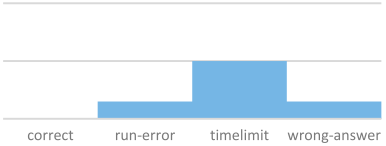
Digital Transmission

Try to put in as many ones as possible, while still keeping the string run-length limited

One more time, but with as few ones as possible

This gives an upper bound and a lower bound on the number of ones, it can be proven that everything in between the bounds can also be constructed

$O(n)$



Gunslinger

Check if possible:

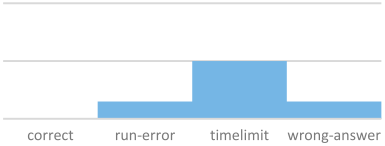
- Compute **convex hull** of all Daltons
- Possible iff Luke and Escape hatch are outside or on the hull

If possible, there are 2 options:

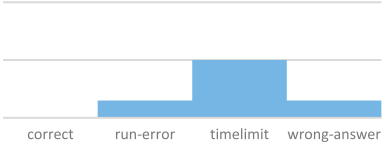
- Direct line between Luke and hatch doesn't intersect hull
- Otherwise, walk via the hull (to implement this, take the hull of all points including Luke and hatch)

$O(n \log n)$

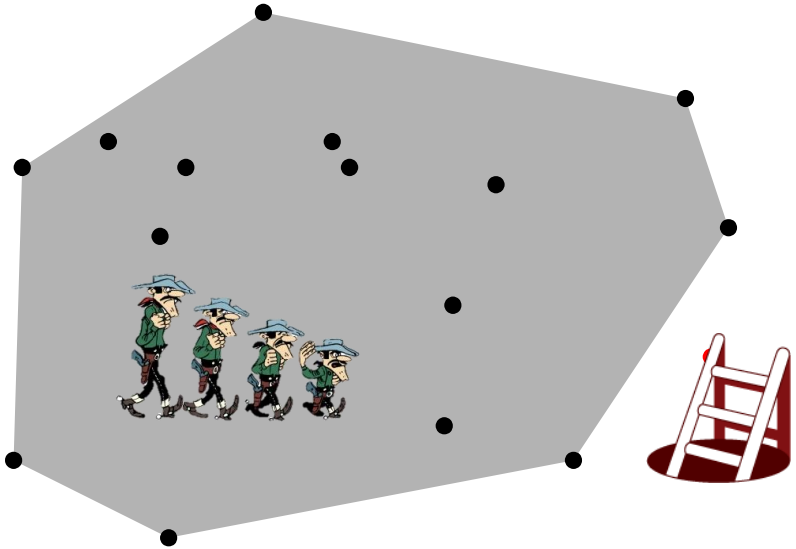
- But $O(n^2)$ was also accepted

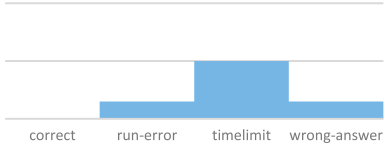


Gunslinger

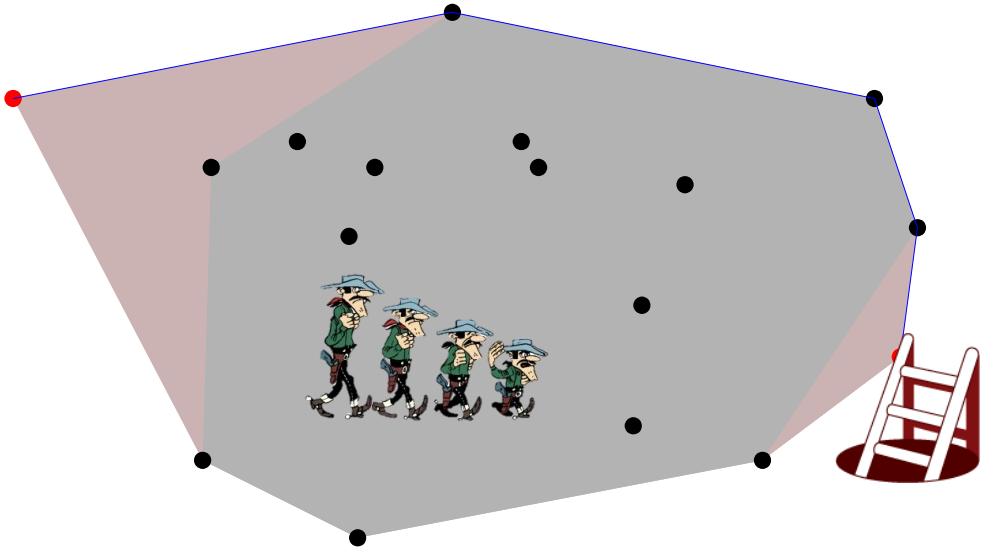


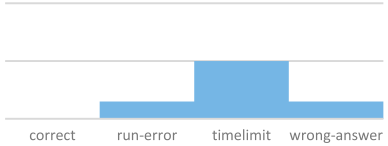
Gunslinger



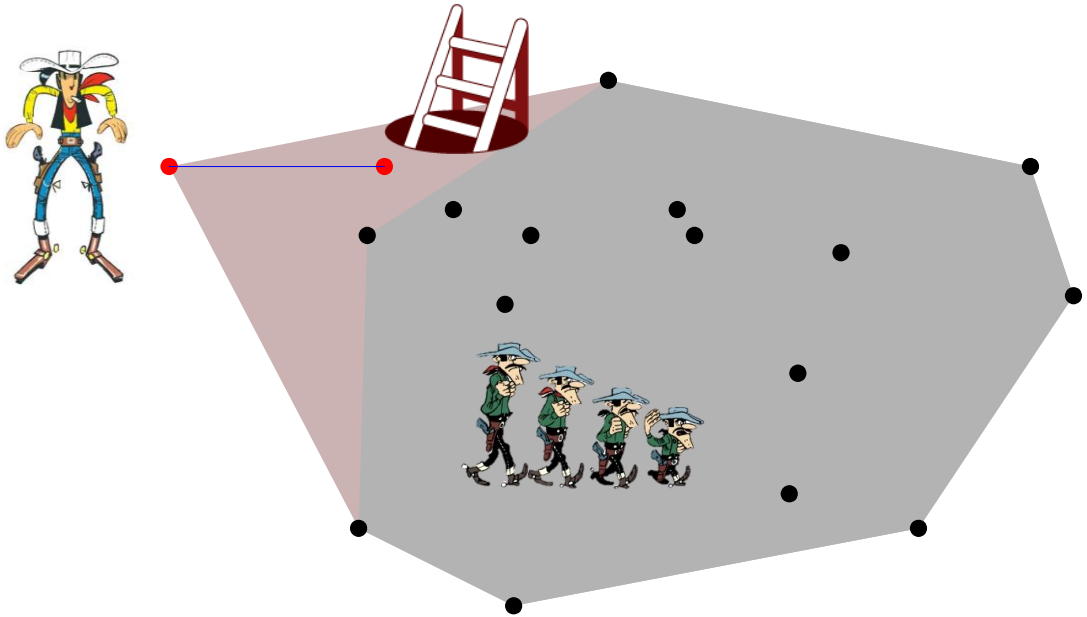


Gunslinger





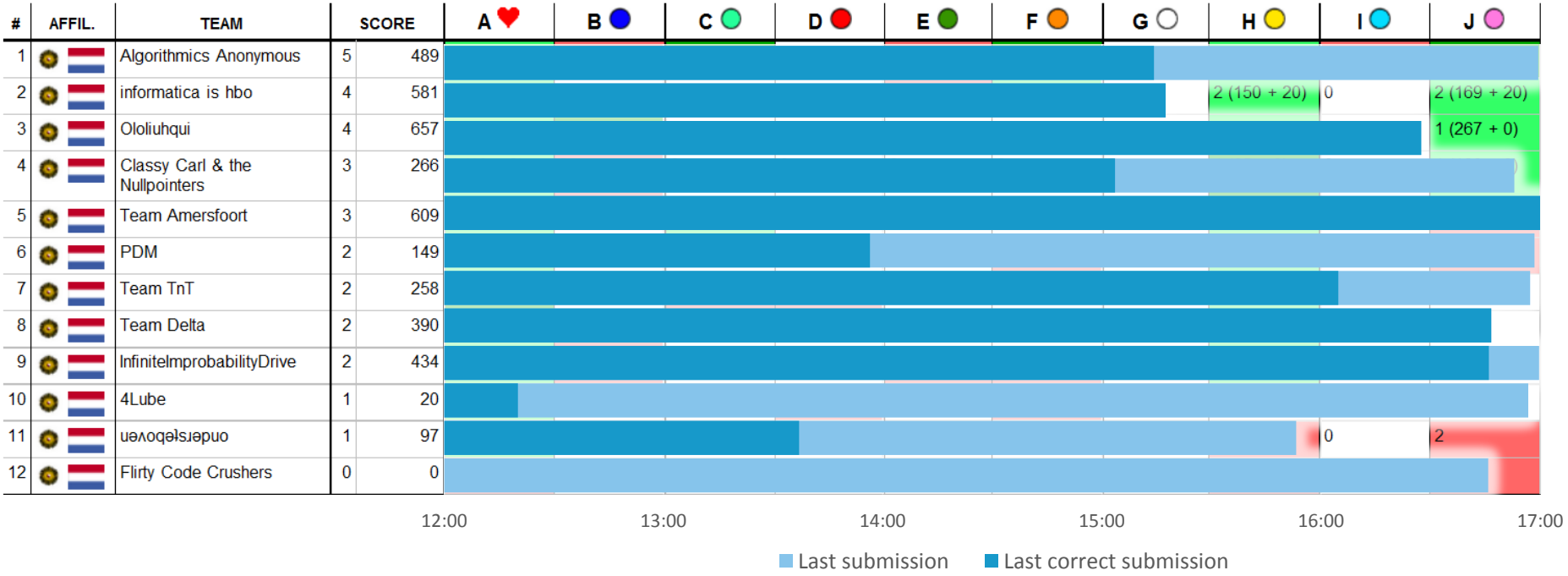
Gunslinger



Scores

programming
contests
are
hot

#	AFFIL.	TEAM	SCORE		A ♥	B ●	C ●	D ●	E ●	F ●	G ○	H ●	I ●	J ●
1		Algorithmics Anonymous	5	489	1 (41 + 0)	3	1 (194 + 0)	0	3	3 (33 + 40)	0	1 (118 + 0)	1	1 (63 + 0)
2		informatica is hbo	4	581	1 (25 + 0)	0	0	0	2	1 (197 + 0)	0	2 (150 + 20)	0	2 (169 + 20)
3		Ololiuhqui	4	657	1 (25 + 0)	2 (141 + 20)	0	0	0	0	0	1 (204 + 0)	0	1 (267 + 0)
4		Classy Carl & the Nullpointers	3	266	1 (18 + 0)	3	0	0	0	0	0	1 (65 + 0)	0	1 (183 + 0)
5		Team Amersfoort	3	609	1 (54 + 0)	0	4 (299 + 60)	0	0	0	0	1 (196 + 0)	0	1
6		PDM	2	149	1 (33 + 0)	0	0	0	0	2	0	1 (116 + 0)	0	1
7		Team TnT	2	258	1 (14 + 0)	0	3	1	0	0	0	1 (244 + 0)	0	0
8		Team Delta	2	390	2 (84 + 20)	6	0	0	0	0	0	1 (286 + 0)	0	0
9		InfiniteImprobabilityDrive	2	434	2 (129 + 20)	2	0	0	1	8	0	1 (285 + 0)	0	0
10		4Lube	1	20	1 (20 + 0)	0	0	2	0	5	0	0	0	0
11		uɐʌoqəlsɹəpuo	1	97	1 (97 + 0)	0	0	0	0	0	0	2	0	2
12		Flirty Code Crushers	0	0	2	0	0	0	0	0	0	4	0	1

When you could’ve gone home

#	AFFIL.	TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
1		Algorithmics Anonymous	5	489										
2		informatica is hbo	4	581								2 (150 + 20)	0	2 (169 + 20)
3		Ololiuhqui	4	657										1 (267 + 0)
4		Classy Carl & the Nullpointers	3	266										
5		Team Amersfoort	3	609										
6		PDM	2	149										
7		Team TnT	2	258										
8		Team Delta	2	390										
9		InfinitelmprobabilityDrive	2	434										
10		4Lube	1	20										
11		uɐʌoqəlsɹəpuo	1	97									0	2
12		Flirty Code Crushers	0	0