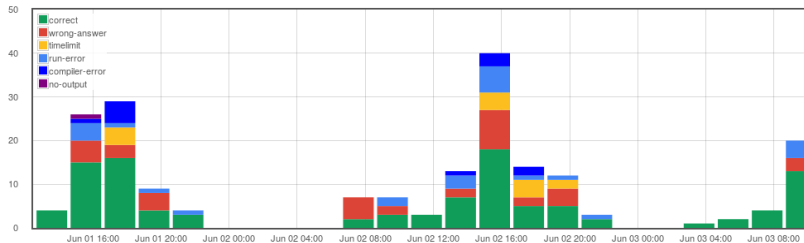


The
German Collegiate Programming Contest 2016
Practice Session

The GCPC 2016 Jury

03.06.2016

Statistics



A: Hello, GCPC! - Sample Solution

Problem

Given a list of numbers, find the largest number x and output x^3 .

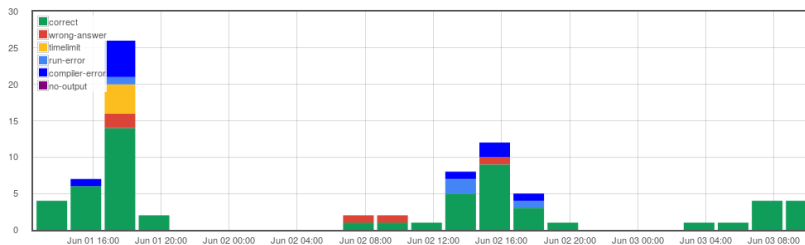
Solution

- ▶ Just do it!
- ▶ Use the long data type!

A: Hello, GCPC! - Statistics

- ▶ Tried by 38 teams (48%), solved by 38 teams (48%)
- ▶ C++: Tried by 20 teams (25%), solved by 20 teams (25%), 57 submissions (35% correct)
- ▶ Java: Tried by 14 teams (18%), solved by 14 teams (18%), 17 submissions (82% correct)
- ▶ Python: Tried by 4 teams (5%), solved by 4 teams (5%), 6 submissions (67% correct)
- ▶ Fastest: 110 minutes, written by PSpace-Orakel
- ▶ Best runtime: 0% of the given time, written by HeapHeapArray, Thinker, Ginke, Danke!, <(OvO)>
- ▶ Shortest: 60 characters, written by The good, the bad & the beauty

A: Hello, GCPC! - Statistics



- ▶ Author: Stefan Toman
- ▶ Keywords: easy - introduction

B: The Buffet Wars - Sample Solution

Problem

Fractional Knapsack Problem: Given some dishes and their reward (noms) per minute, decide how much of each dish you should eat.

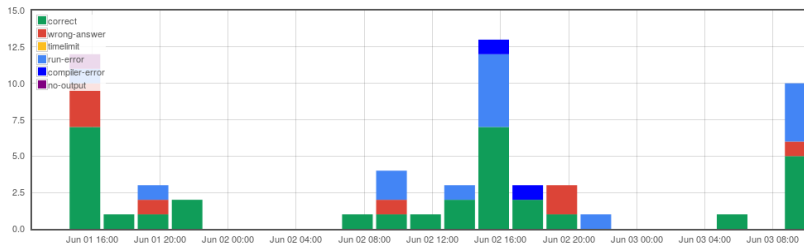
Solution

- ▶ Always pick the dish with the most noms per serving.
- ▶ Do so until either
 - ▶ the maximum amount of servings for the dish is reached or
 - ▶ Herberts plate is full.

B: The Buffet Wars - Statistics

- ▶ Tried by 33 teams (41%), solved by 29 teams (36%)
- ▶ C++: Tried by 17 teams (21%), solved by 15 teams (19%), 28 submissions (54% correct)
- ▶ Java: Tried by 14 teams (18%), solved by 13 teams (16%), 27 submissions (48% correct)
- ▶ Python: Tried by 2 teams (3%), solved by 1 teams (1%), 3 submissions (33% correct)
- ▶ Fastest: 127 minutes, written by oachkatzlschwoaf
- ▶ Best runtime: 0% of the given time, written by 14 teams
- ▶ Shortest: 373 characters, written by Snow White and the Huntsmen

B: The Buffet Wars - Statistics



- ▶ Author: Christian Müller
- ▶ Keywords: easy - Greedy

C: Packing Cases - Sample Solution

Problem

Decide how high you can stack boxes of certain sizes, but you can only stack smaller boxes on larger ones.

Insight

- ▶ Each box may only be used two times \Rightarrow always enough boxes.

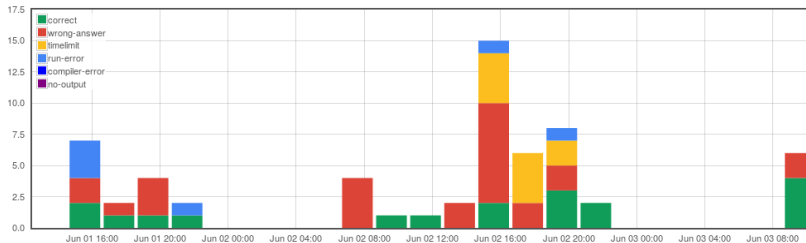
Solution

- ▶ Add each rotation of each box (six/three if properly sorted).
- ▶ Sort the boxes by decreasing area on the top, then use DP.
- ▶ $a[i]$: maximum height using boxes 1 to i with box i on top.
- ▶ $a[i] = \max_{1 \leq j < i} \{a[j] + h_i\}$ if box i fits on top of box j .
- ▶ Return $\max_i a[i] \geq h$.

C: Packing Cases - Statistics

- ▶ Tried by 19 teams (24%), solved by 13 teams (16%)
- ▶ C++: Tried by 11 teams (14%), solved by 9 teams (11%), 35 submissions (26% correct)
- ▶ Java: Tried by 8 teams (10%), solved by 4 teams (5%), 25 submissions (16% correct)
- ▶ Python: Tried by 0 teams (0%), solved by 0 teams (0%), 0 submissions (0% correct)
- ▶ Fastest: 153 minutes, written by oachkatzlschwoaf
- ▶ Best runtime: 1% of the given time, written by Platz 56 oder disqualifiziert, dfg, <(OvO)>, Thinke, Ginke, Danke!, The SegFAULT in Our Stars
- ▶ Shortest: 847 characters, written by oachkatzlschwoaf

C: Packing Cases - Statistics



- ▶ Author: Philipp Hoffmann
- ▶ Keywords: medium - Dynamic Programming

GCPC 2016

See you tomorrow!

Be on time, we will begin with the announcements at 11:30 a.m.