

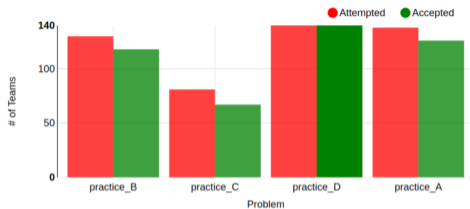
Practice Problem Analysis Session

SWERC judges

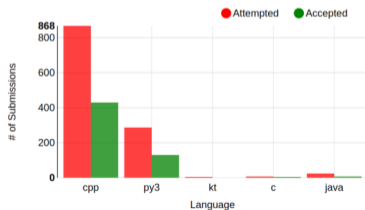
30/11/2024

Statistics

Number of submissions: about 1194

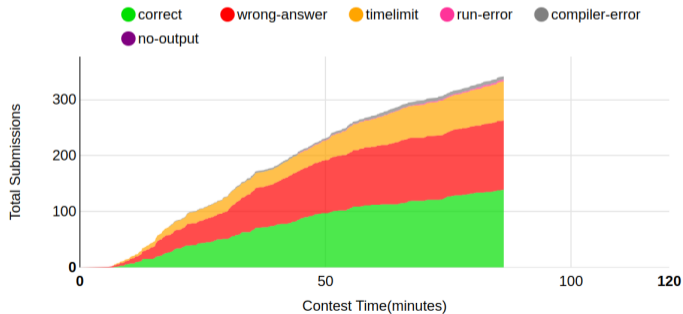


Number of clarification requests: 145 (about 10 answered “No comment.”)



A: Find the Tomb

Solved by 101 teams before freeze.
First solved after 6 min by **Artois University**.



A: Find the Tomb

Problem

Find a value between 1 and 1000 by bisection.

Solution

Interactive problem. Bisect the answer: keep a [high, low] range and ask about the middle of the range, then repeat. Read each answer on stdin and output each guess on stdout.

A: Find the Tomb

Example code

```
left = 1
right = 1000

while True:
    middle = (left + right) // 2
    print(middle, flush=True)
    response = input().strip()
    if response == "found":
        break
    elif response == "solid":
        right = middle - 1
    else: # response == "hollow"
        left = middle + 1
```

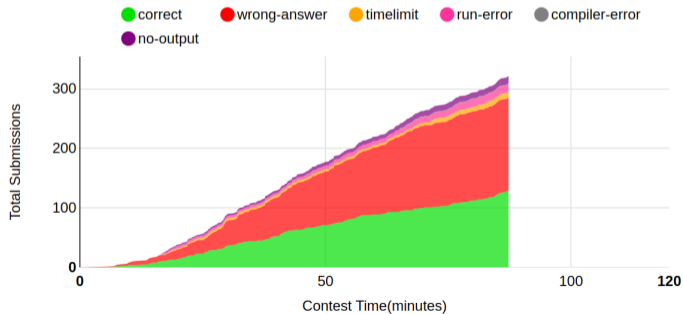
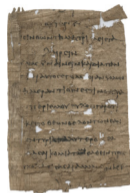
A: Find the Tomb

Example code

```
while (true) {  
    int middle = (left + right) / 2;  
    cout<<middle<<endl<<flush;  
    string response;  
    cin>>response;  
    if (response == "found") {  
        break;  
    } else if (response == "solid") {  
        right = middle - 1;  
    } else {  
        assert(response == "hollow");  
        left = middle + 1;  
    }  
}
```

B: Etymology

Solved by 16 teams before freeze.
First solved after 6 min by **Ecole Polytechnique**.



B: Etymology

Problem

Find a string with length L , such that it's the prefix of at least M words.

Solution 1 – $O(N^2 * L)$ time

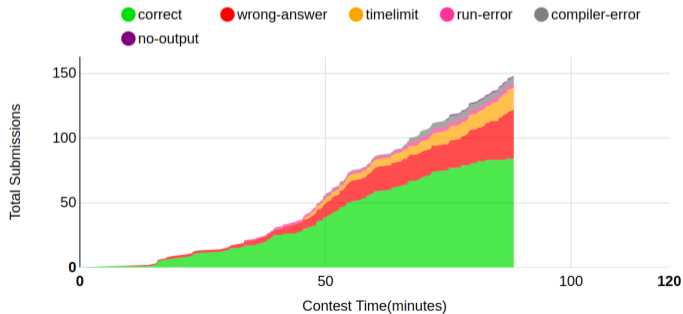
Compare each word with all other words, and see if its prefix (with length L) is also the prefix of at least other $M - 1$ words.

Solution 2 – $O(N * L)$ time

Use a map with key as prefix and value as the number of occurrences. Add the words into the map one by one, and see if there's a prefix with at least M occurrences.

C: Election

Solved by 106 teams before freeze.
First solved after 7 min by
Institut Polytechnique de Paris.



C: Election

Problem

In a winner-take-all system, given the probability of a candidate winning in each centuria, calculate the probability of winning the whole election.

Solution – $O(N * \text{total_votes})$ time

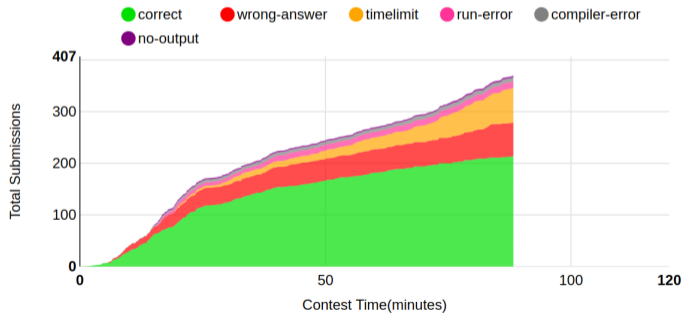
- Dynamic programming. Let $f[i][j]$ be the probability of getting exactly j votes in the first i centuriae. Then we have:

$$f[i][j] = f[i - 1][j - w_i] \cdot p_i + f[i - 1][j] \cdot (1 - p_i)$$

- Finally, the overall probability is $\sum_{j=\lceil \text{total_votes}/2 \rceil}^{\text{total_votes}} f[N][j]$.

D: Help The Organizers

Solved by 20 teams before freeze.
First solved after 1 min by **Ecole Polytechnique**.



D: Help The Organizers

Problem

Find the number of hosts needed for $t_1 + t_2 + t_3$ teams. A host can serve at most 20 teams.

Solution

The solution is $(t_1 + t_2 + t_3 + 19) // 20$ where $//$ represents integer division.