

Name: Sagar Srivastava

Roll No: 1900290100127

LeetCode Weekly Contest 301

Week 1

In this contest, I solved 3/4 questions and got a rank of 3433 / 23566

Contest - LeetCode

← → ↻ leetcode.com/contest/weekly-contest-301/

Google YouTube Aptitude Technical MCQ's DSA Sheet (1) Dynamic Progra... Trie Series | Striver ... Master Graphs OS, DBMS, CN - Stri... Project Explain OOPs You.com | The searc... Java Q's

Announcement
Users **must register** to participate. We hope you enjoy this contest!

Do you enjoy the contest? Click [HERE](#) to fill out a quick survey and help LeetCode provide a better contest experience. "Hey LeetCodeers! If you have participated in this contest, you might be able to win a small prize! Fill out this survey before Sunday, 12pm PST for a chance to be one of 10 lucky contestants to win 10 LeetCodeCoins!"

Want millions of LeetCode users to *recognize your company*? [Contact us](#) to sponsor a contest.

[Quit Virtual Contest](#) [FAQ about Contest](#)

Problem list **Score**

Minimum Amount of Time to Fill Cups	3
Smallest Number in Infinite Set	4
Move Pieces to Obtain a String	5
Count the Number of Ideal Arrays	6

Ranking **Score**

Rank	Name	Score	Finish Time
1	lzqshmlly	0	0:00:00
2	vldmr256	0	0:00:00
3	kkg2002	0	0:00:00
4	bdiptoroy	0	0:00:00
5	hope_ma	0	0:00:00

More...

Type here to search

30°C 01:38 10/8/2022

The questions that I've solved are below:

- 1) Minimum Amount of Time to Fill Cups
- 2) Smallest Number in Infinite Set
- 3) Move Pieces to obtain a string

1) Minimum Amount of Time to Fill Cups

```
class Solution {
public:
    int fillCups(vector<int>& amount) {
        priority_queue<int> pq;
        for(auto x : amount){
            if(x != 0){
                pq.push(x);
            }
        }
        int ans = 0;

        while(pq.size() > 0){
            if(pq.size() == 1){
                int x = pq.top();
                pq.pop();
                ans += x;
            }
            else{
                int a = pq.top();
                pq.pop();

                int b = pq.top();
                pq.pop();

                a--;
                b--;

                if(a > 0){
```

```

        pq.push(a);
    }

    if(b > 0){
        pq.push(b);
    }

    ans++;
}
}
return ans;
}
};

```

2) Smallest Number in Infinite Set

```

class SmallestInfiniteSet {
public:
    set<int> s;
    SmallestInfiniteSet() {
        s.clear();
        for(int i=1;i<=1000;i++){
            s.insert(i);
        }
    }

    int popSmallest() {
        int smallest = *s.begin();
        s.erase(smallest);
        return smallest;
    }
}

```

```

void addBack(int num) {
    s.insert(num);
}
};

```

3) Move Pieces to Obtain a String

```

class Solution {
public:
    bool canChange(string start, string target) {
        int n = target.length();
        int m = start.length();

        int a = 0, b = 0, i = 0, j = 0;
        while(i < n and j < m){
            while(i < n and target[i] == '_') i++;
            while(j < m and start[j] == '_') j++;

            if(i == n || j == m){
                return i == n and j == m;
            }

            if(target[i] != start[j]) return false;

            if(target[i] == 'L'){
                if(j < i) return false;
            }
            else{
                if(i < j) return false;
            }
        }
    }
};

```

```
        i++;
        j++;
    }

    while(i < n and target[i] == '_') i++;
    while(j < n and start[j] == '_') j++;
    return i == n and j == n;
}

};
```