

Advanced Programming

ASSIGNMENT 08

Q1. Max Units on a Truck.

Code:

```
</> Code | ☒ Testcase | >_ Test Result

C++ v  Auto  

1  class Solution {
2  public:
3      int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
4          for (int i=0;i<boxTypes.size();i++){
5              reverse(boxTypes[i].begin(),boxTypes[i].end());
6          }
7          sort(boxTypes.begin(),boxTypes.end());
8          reverse(boxTypes.begin(),boxTypes.end());
9          int ans=0,i=0;
10         while (truckSize>0 && i<boxTypes.size()){
11             while (truckSize>0 && boxTypes[i][1]>0){
12                 ans+=boxTypes[i][0];
13                 truckSize--;
14                 boxTypes[i][1]--;
15             }
16             i++;
17         }
18         return ans;
19     }
20 };
```

Output:

Description | Editorial | Submissions | Solutions | Accepted ×

← All Submissions

Accepted 76 / 76 testcases passed

saloni_pundir_ submitted at Apr 03, 2025 18:44

Editorial Solution

Runtime

13 ms | Beats 24.31%

Analyze Complexity

Memory

19.91 MB | Beats 50.88%

</> Code | ☒ Testcase | >_ Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

boxTypes =
[[1,3],[2,2],[3,1]]

truckSize =
4

Output

8

Expected

8

Q2. Min Operations to Make Array Increasing.

Code:

```
</> Code | ☒ Testcase | >_ Test Result

C++ ☐ Auto

1  class Solution {
2  public:
3      int minOperations(vector<int>& nums) {
4          int output=0;
5          for(int i=0;i<nums.size()-1;i++){
6              if(nums[i]<nums[i+1])
7                  continue;
8              else{
9                  output=output+(nums[i]+1-nums[i+1]);
10                 nums[i+1]=nums[i]+1;
11             }
12         }
13         return output;
14     }
15 };
```

Output:

Description | Editorial | Submissions | Solutions | Accepted X

< All Submissions

Accepted 94 / 94 testcases passed

saloni_pundir_ submitted at Apr 03, 2025 18:46

Solution

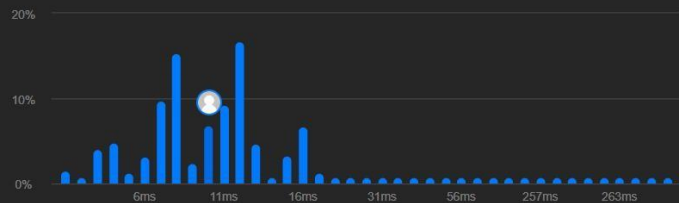
Runtime

10 ms | Beats 57.73%

Analyze Complexity

Memory

19.67 MB | Beats 28.34%



</> Code | ☒ Testcase | >_ Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

nums =
[1,1,1]

Output

3

Expected

3

Q3. Remove Stones to Maximize Total.

Code:

```
</> Code | Testcase | Test Result
C++ v Auto

1 class Solution {
2 public:
3 bool static help(int x,int y)
4 {
5     return x>y;
6 }
7 int minStoneSum(vector<int>& piles, int k) {
8
9     int n=piles.size();
10    priority_queue<int,vector<int>> pq(piles.begin(),piles.end());
11    int ans=accumulate(piles.begin(),piles.end(),0);
12    int i=0;
13    while(k>0 && !pq.empty())
14    {
15        int temp=pq.top();
16        pq.pop();
17        ans-=(temp/2);
18        pq.push(temp-temp/2);
19        k--;
20    }
21
22    return ans;
23 }
24 };
```

Output:

Description | Editorial | Submissions | Solutions | Accepted x

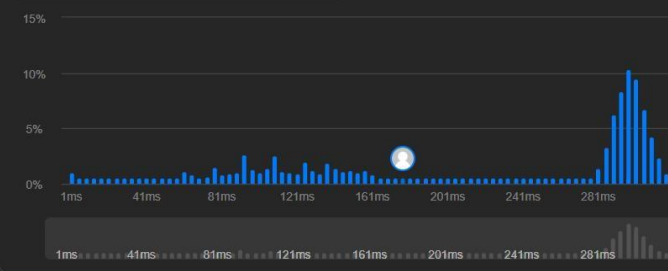
← All Submissions

Accepted 60 / 60 testcases passed
saloni_pundir_ submitted at Apr 03, 2025 18:48

Editorial Solution

Runtime
180 ms | Beats 97.18%
Analyze Complexity

Memory
102.82 MB | Beats 77.26%



Code | C++

</> Code | Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input
piles =
[5,4,9]
k =
2

Output
12

Expected
12

Con

Q4. Max Score from Removing Substrings.

Code:

```
Code | Testcase | Test Result
C++ v Auto
1 class Solution {
2 public:
3     int maximumGain(string s, int x, int y) {
4         vector<char> v, v1;
5         int ans = 0;
6         for (auto a : s) {
7             if (!v.empty()) {
8                 if (x > y) {
9                     if (v.back() == 'a' && a == 'b') {
10                         ans += x;
11                         v.pop_back();
12                     } else {
13                         v.push_back(a);
14                     }
15                 } else {
16                     if (v.back() == 'b' && a == 'a') {
17                         ans += y;
18                         v.pop_back();
19                     } else {
20                         v.push_back(a);
21                     }
22                 }
23             } else {
24                 v.push_back(a);
25             }
26         }
27         for (auto a : v) {
28             if (!v1.empty()) {
29                 if (x < y) {
30                     if (v1.back() == 'a' && a == 'b') {
31                         ans += x;
32                         v1.pop_back();
33                     } else {
34                         v1.push_back(a);
35                     }
36                 } else {
37                     if (v1.back() == 'b' && a == 'a') {
38                         ans += y;
39                         v1.pop_back();
40                     } else {
41                         v1.push_back(a);
42                     }
43                 }
44             } else {
45                 v1.push_back(a);
46             }
47         }
48         return ans;
49     }
50 }
```

Output:

Description | Editorial | Submissions | Solutions | Accepted x

← All Submissions

Accepted 77 / 77 testcases passed

saloni_pundir_ submitted at Apr 03, 2025 18:50

Editorial Solution

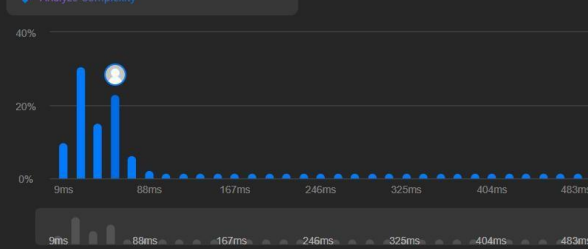
Runtime

62 ms | Beats 29.54%

Analyze Complexity

Memory

24.42 MB | Beats 75.07%



Code | Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

S =

"cdbcbbaaabab"

x =

4

y =

5

Output

19

Expected

19

Q5. Min Operations to Make a Subsequence.

Code:

```
C++ v Auto
1  class Solution {
2  public:
3      int minOperations(vector<int>& target, vector<int>& arr) {
4          unordered_map<int, int> mp;
5          for(int i = 0; i < target.size(); i++) mp[target[i]] = i;
6          vector<int> v;
7          for(int a: arr) if (mp.count(a)) v.push_back(mp[a]);
8          int n = v.size(), ans = 0;
9          vector<int> tail(n + 1, INT_MAX);
10         tail[0] = INT_MIN;
11         for(int a: v) {
12             int b = upper_bound(tail.begin(), tail.begin() + min(ans + 1, n), a) - tail.begin();
13             if (b == 0 || (tail[b - 1] < a && tail[b] > a)) {
14                 tail[b] = a;
15                 ans = max(ans, b);
16             }
17         }
18         return target.size() - ans;
19     }
20 ;
```

Output:

