



















All Competitions > University CodeSprint 4 > Cubes and Cylinders

## **Cubes and Cylinders**



ted a few sec	conds ago • Score: 30.00				Status:	Accepted
~	Test Case #0	~	Test Case #1	~	Test Case #2	
~	Test Case #3	<b>~</b>	Test Case #4	<b>~</b>	Test Case #5	
~	Test Case #6	<b>~</b>	Test Case #7	<b>✓</b>	Test Case #8	
~	Test Case #9	<b>✓</b>	Test Case #10	<b>~</b>	Test Case #11	
~	Test Case #12	<b>✓</b>	Test Case #13	<b>~</b>	Test Case #14	
~	Test Case #15	<b>~</b>	Test Case #16	<b>~</b>	Test Case #17	
~	Test Case #18	<b>~</b>	Test Case #19	<b>~</b>	Test Case #20	
•	Test Case #21	<b>~</b>	Test Case #22	<b>~</b>	Test Case #23	
~	Test Case #24					

## **Submissions by Friends**

Rank	User	Score	School
1	NiceBuddy	25.242	Universität Stuttgart

Full Contest Leaderboard

## **Submitted Code**

```
Language: C++14
                                                                                                          Open in editor
 1 #include <bits/stdc++.h>
 3 using namespace std;
 4 const float root = sqrt(2);
 6 inline void swap(int *xp, int *yp)
 8
       int temp = *xp;
 9
       *xp = *yp;
10
       *yp = temp;
11 }
13 inline void bubbleSort(vector<int> &S, vector<int> &K, unsigned int n)
14 {
15
       int i, j;
16
       for (i = 0; i < n-1; i++)
           for (j = 0; j < n-i-1; j++)
17
18
               if (S[j] > S[j+1])
19
```

```
20
                      swap(&S[j], &S[j+1]);
21
                      swap(&K[j], &K[j+1]);
22
                  }
23 }
24 vector<string> split_string(string);/** n
                                                                                         m */
                                                         n
25 inline int maximumPackages(vector<int> &S, vector<int> &K, vector<int> &R, vector<int> &C)
26 {
27
      int count = 0;
28
      float LHS = 0.0;
      float RHS = 0.0;
29
30
      bubbleSort(S,K, S.size());
31
32
      bubbleSort(R,C, R.size());
33
34
      for(unsigned int i = 0; i<S.size(); ++i)</pre>
35
36
          int curEdge = S[i];
37
          int currNo = K[i];
38
39
          for(unsigned int j = 0; j<R.size(); ++j)</pre>
40
41
              int currRadius = R[j];
42
              int currCapa
                            = C[j];
43
              if( (C[j] !=0 ) && (currNo != 0 ) )
44
45
46
                  LHS = static_cast<float>(currRadius)*2;
                  RHS = static_cast<float>(curEdge)*root;
47
48
                  if(LHS>RHS)
49
50
                  {
51
                      if(currNo > currCapa)
52
                      {
53
                          count+=currCapa;
54
                          C[j] = 0;
55
                          currNo = currNo - currCapa;
56
                      }
57
                      else
                              // currNo < currCapa
58
                      {
59
                          count+=currNo:
60
                          C[j] = C[j] - currNo;
61
                          currNo = 0;
62
                      }
                  }
63
64
              }
65
          }
66
      }
67
68
      return count;
69 }
70
71
72 int main()
73 {
74
      ios_base::sync_with_stdio(false);
75
      cin.tie(nullptr);
                          cout.tie(nullptr);
76
77
      string nm_temp;
78
      getline(cin, nm_temp);
79
80
      vector<string> nm = split_string(nm_temp);
81
      int n = stoi(nm[0]);
      int m = stoi(nm[1]);
82
83
84
       85
      string S_str_temp;
                             getline(cin, S_str_temp);
      vector<string> S_str = split_string(S_str_temp);
86
87
      vector<int> S(n);
88
      for (int S_i = 0; S_i < n; S_{i++})
89
      {
90
          int S_item = stoi(S_str[S_i]);
91
          S[S_i] = S_item;
92
       93
94
95
       string K_str_temp;
                              getline(cin, K_str_temp);
96
      vector<string> K_str = split_string(K_str_temp);
97
      vector<int> K(n);
```

```
2/24/2018
```

```
98
       for (int K_i = 0; K_i < n; K_{i++})
99
           int K_item = stoi(K_str[K_i]);
100
101
           K[K_i] = K_item;
102
       103
104
105
       //cin.ignore(numeric_limits<streamsize>::max(), '\n');
106
       string R_str_temp;
                            getline(cin, R_str_temp);
       vector<string> R_str = split_string(R_str_temp);
107
108
       vector<int> R(m);
       for (int R_i = 0; R_i < m; R_{i++})
109
110
       {
111
           int R_item = stoi(R_str[R_i]);
112
           R[R_i] = R_item;
113
       114
115
116
       string C_str_temp;
                             getline(cin, C_str_temp);
117
       vector<string> C_str = split_string(C_str_temp);
       vector<int> C(m);
118
119
       for (int C_i = 0; C_i < m; C_{i++})
120
121
           int C_item = stoi(C_str[C_i]);
122
           C[C_i] = C_item;
123
124
       125
126
       int result = maximumPackages(S, K, R, C);
127
       cout << result << "\n";</pre>
128
129
130
       return 0:
131 }
132
133 inline vector<string> split_string(string input_string)
134 {
135
       string::iterator new_end = unique(input_string.begin(), input_string.end(), [] (const char &x, const char &y) {
           return x == y and x == ' ';
136
137
       }):
138
139
       input string.erase(new end, input string.end());
140
       while (input_string[input_string.length() - 1] == ' ') {
141
142
           input_string.pop_back();
143
144
145
       vector<string> splits;
       char delimiter = ' ';
146
147
148
       size t i = 0:
149
       size_t pos = input_string.find(delimiter);
150
151
       while (pos != string::npos) {
152
           splits.push_back(input_string.substr(i, pos - i));
153
154
           i = pos + 1;
           pos = input_string.find(delimiter, i);
155
156
157
       splits.push_back(input_string.substr(i, min(pos, input_string.length()) - i + 1));
158
159
       return splits;
160
161 }
162
```

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature