hackerearth

Platform jumps / Submission (62263652) by Suraj Kumar (ThinkTwice)

Submission ID: 62263652 / 5 hours ago

**RESULT:** Accepted — Refer judge environment

| Time (sec) | Memory (KiB) | Language |
|---|---|---|
| 1.42648 | 7440 | C++17 |

| Visibility | Input | Result | Time (sec) | Memory (KiB) | Your Output | Correct Output |
|---|---|---|---|---|---|---|
| | Input #1 | Accepted | 0.009289 | 2 | - | - |
| | Input #2 | Accepted | 0.009048 | 2 | - | - |
| | Input #3 | Accepted | 0.008939 | 2 | - | - |
| | Input #4 | Accepted | 0.009776 | 2 | - | - |
| | Input #5 | Accepted | 0.009136 | 2 | - | - |
| | Input #6 | Accepted | 0.295667 | 7420 | - | - |
| | Input #7 | Accepted | 0.247969 | 7440 | - | - |
| | Input #8 | Accepted | 0.237821 | 7352 | - | - |
| | Input #9 | Accepted | 0.304582 | 7320 | - | - |
| | Input #10 | Accepted | 0.294253 | 7228 | - | - |

**Submission:**

Theme: Light

```cpp
#include<bits/stdc++.h>
using namespace std;

void bfs(int node, vector <vector <int>> &graph, vector <int> &res, int curr) {
    queue <int> q;
    q.push(node);
    res[node] = curr;
    while(q.size()) {
        node = q.front();
        q.pop();
        for(auto child: graph[node]) {
            if(res[child] == -1) {
                res[child] = res[node] + 1;
                q.push(child);
            }
        }
    }
}

vector<int> Min_Jumps (int n, vector<int> a) {
    // Write your code here
    vector <int> ngr, ngl;
    stack <int> s;
    for(int i = 0; i<n; i++) {
        while(s.size() && a[s.top()] < a[i]) s.pop();
        if(s.size() == 0) ngl.push_back(-1);
        else ngl.push_back(s.top());
        s.push(i);
    }
    while(s.size()) s.pop();
    for(int i = n-1; i>=0; i--) {
        while(s.size() && a[s.top()] < a[i]) s.pop();
        if(s.size() == 0) ngr.push_back(-1);
        else ngr.push_back(s.top());
        s.push(i);
```

```
    }
    reverse(ngr.begin(), ngr.end());
    int ind = -1;
    int mx = -1;
    for(int i = 0; i<n; i++) {
        if(a[i] > mx) {
            mx = a[i];
            ind = i;
        }
    }
    ngr[ind] = ind;
    ngl[ind] = ind;

    vector <vector <int>> graph(n+1);
    for(int i = 0; i<n; i++) {
        if(ngr[i] != -1) {
            graph[ngr[i]].push_back(i);
        }
    }
    for(int i = 0; i<n; i++) {
        if(ngl[i] != -1) {
            graph[ngl[i]].push_back(i);
        }
    }

    // for(int i = 0; i<n; i++) {
    //     cout<<i<<" -> ";
    //     for(auto x: graph[i]) cout<<x<<" ";
    //     cout<<endl;
    // }
    // cout<<endl;
    // for(auto x: ngr) cout<<x<<" ";
    // cout<<endl;
    // for(auto x: ngl) cout<<x<<" ";
    // cout<<endl;
    vector <int> res(n, -1);
    bfs(ind, graph, res, 0);

    return res;
}

int main() {
```

```
78.
79.       ios::sync_with_stdio(0);
80.       cin.tie(0);
81.       int T;
82.       cin >> T;
83.       for(int t_i = 0; t_i < T; t_i++)
84.       {
85.           int N;
86.           cin >> N;
87.           vector<int> A(N);
88.           for(int i_A = 0; i_A < N; i_A++)
89.           {
90.                   cin >> A[i_A];
91.           }
92.
93.           vector<int> out_;
94.           out_ = Min_Jumps(N, A);
95.           cout << out_[0];
96.           for(int i_out_ = 1; i_out_ < out_.size(); i_out_++)
97.           {
98.                   cout << " " << out_[i_out_];
99.           }
100.          cout << "\n";
101.      }
102.  }
```

**Embed Submission:**

<iframe src=https://www.hackerearth.com/submission/key/a9961093baa34b8b84f2d7c65025d8a3/?theme=light width='100%' height='1875px' frameborder='0' allowtransparency='true' scrolling='yes'></iframe>

Theme: Light

COMMENTS (0)

Start Discussion...

Cancel Post

+1-650-461-4192

contact@hackerearth.com

**Resources**

- Tech Recruitment Blog
- Product Guides
- Developer hiring guide
- Engineering Blog
- Developers Blog
- Developers Wiki
- Competitive Programming
- Start a Programming Club
- Practice Machine Learning

**Solutions**

- Assess Developers
- Conduct Remote Interviews
- Assess University Talent
- Organize Hackathons

**Company**

- About Us
- Press
- Careers

**Service & Support**

- Technical Support
- Contact Us