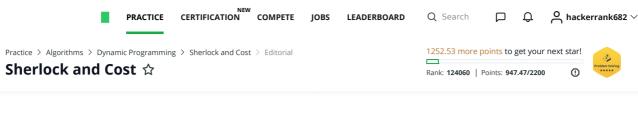
Medium 52.02%

O(N)

#include <cmath>



```
Problem
                      Submissions
                                            Leaderboard
                                                                   Discussions
                                                                                         Editorial
                                                                                                                        STATISTICS
LK Editorial by Lalit Kundu
Problem: Array \pmb{A} contains the elements \pmb{A_1,A_2\ldots A_N}. And array \pmb{B} contains the elements \pmb{B_1,B_2\ldots B_N}. There is a
                                                                                                                        Difficulty:
relationship between A_i and B_i orall 1 \leq i \leq N. That is, for any valid i, 1 \leq A_i \leq B_i.
                                                                                                                        Success Rate:
                                                                                                                        Time Complexity:
The cost m{S} of m{A} is defined as:
                                                                                                                        Required Knowledge:
                                             S = \sum_{i=2}^N |A_i - A_{i-1}|
                                                                                                                                                     May 30 2014
                                                                                                                        Publish Date:
                                                                                                                        Originally featured in 101 Hack May
Print the largest possible value of \boldsymbol{S}.
                                                                                                                        Of the 1164 contest participants, 198 (17.01%)
Solution: For S to be largest, A_i will either be 1 or B_i because |1-x|>(|2-x|=|1-(x-1)|). We can make a
                                                                                                                        submitted code for this challenge.
simple DP now, say DP[n][2]. DP[i][0] stores the maximum value of S using the first i elements only if A_i was 1. If
m{A[i]} = m{B[i]}, m{DP[i]}[1] stores the maximum value of m{S} possible using the first m{i} elements only.
                                                                                                                        NEED HELP?
See setter's solution for more details.
                                                                                                                        View discussions
LK Set by Lalit Kundu

▼ View top submissions

Problem Setter's code:
C++
   #include<bits/stdc++.h>
  using namespace std;
   int ar[100005]={},dp[100005][2]={};
   int main()
        int t;
       cin >> t;
       while(t--)
            int n,i,j;
            cin >> n;
            for(i=0; i<n; i++)
                 cin >> ar[i]:
            for(i=0; i<n-1; i++)
                 dp[i+1][0]=max(dp[i][0],dp[i][1]+abs(ar[i]-1));
                 dp[i+1][1]=max(dp[i][0]+abs(ar[i+1]-1),dp[i][1]+abs(ar[i]-ar[i+1]));
            cout << max(dp[n-1][0],dp[n-1][1]) << endl;
       return 0;
Tested by gera1d
Problem Tester's code:
C++
  #ifdef ssul
  \#define \_GLIBCXX\_DEBUG
   #endif
  #undef NDEBUG
  #include <algorithm>
  #include <functional>
  #include <numeric>
  #include <iostream>
  #include <cstdio>
```

11/24/20, 2:55 PM 1 of 3

```
#include <cstdlib>
#include <ctime>
#include <cstring>
#include <cassert>
#include <vector>
#include <list>
#include <map>
#include <set>
#include <deque>
#include <queue>
#include <bitset>
#include <sstream>
using namespace std;
#define fore(i, l, r) for(int i = (l); i < (r); ++i)
#define forn(i, n) fore(i, 0, n)
#define fori(i, l, r) fore(i, l, (r) + 1)
#define sz(v) int((v).size())
#define all(v) (v).begin(), (v).end()
#define pb push back
#define mp make_pair
#define X first
#define Y second
#if ( _WIN32 || __WIN32__ )
#define LLD "%I64d"
#else
   #define LLD "%lld"
#endif
typedef long long li;
typedef long double ld;
typedef pair<int, int> pt;
template<typename T> T abs(T a) { return a < 0 ? -a : a; }</pre>
template<typename T> T sqr(T a) { return a*a; }
const int INF = (int)1e9;
const ld EPS = 1e-9;
const ld PI = 3.1415926535897932384626433832795;
int readInt(int l, int r){
    if(scanf("%d", &x) != 1){
        fprintf(stderr, "Expected int in range [%d, %d], but haven't found!", l, r);
    if(!(l <= x && x <= r)){
        fprintf(stderr, "Expected int in range [%d, %d], but found %d!", l, r, x);
        throw:
    return x;
}
int up[2], nup[2];
void solve(int tid){
    int n = readInt(1, 100000);
    vector<int> b(n);
    forn(i, n){
        b[i] = readInt(1, 100) - 1;
    }
    forn(i, 2)
        up[i] = 0;
    fore(i, 1, n){
        memset(nup, 0, sizeof nup);
        forn(j, 2){
                nup[j] = max(nup[j], up[k] + abs((k ? b[i - 1] : 0) - (j ? b[i] : 0)));
        memcpy(up, nup, sizeof nup);
    cout << max(up[0], up[1]) << endl;</pre>
#ifdef ssul
    assert(freopen("input.txt", "rt", stdin));
```

2 of 3 11/24/20, 2:55 PM

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

3 of 3 11/24/20, 2:55 PM