b'

Google Software Engineering Intern, Fall 2019 \xe2\x80\x93 North America

- Difficulty Level :\nEasy
- Last Updated :\n03 Jul, 2019

1. Consider a binary tree of N vertices such that children of node k are 2*k and 2*k+1. Vertex 1 is the root of the tree and each node has an integer value associated with it.

Such a tree may be represented as an array of N integers by writing down values from consecutive nodes.

The tree can be represented as an array [-1, 7, 0, 7, -8].
A node is said to be at level x if the length of the shortest path between that node and root x-1. So, the root is at level 1, the children of root are at level 2, and so on.

Your task is to find the smallest level number x such that sum of all nodes at level x is maximal.
Examples: Given array A such that: A[0]=-1, A[1]=7, A[2]=0, A[3]=7, A[4]=-8. The function should return 2.

Input : [-1, 7, 0, 7, -8] \r\n**Output : 2**\r\n

```
#include <iostream>
using namespace std;
int solution(int a[], int n)
{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int max = -1;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int temp = 0;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0for (int i = 0; i < n; i = i + 2) {
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0if (i == 0)
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0temp = a[i];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0else
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0temp = a[i] +
a[i - 1];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0if (temp > max)
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0max = i;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0return max;
}
\xc2\xa0\xc2\xa0
int main()
{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int a[4];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0a[0] = -1, a[1] = 7, a[2] = 0, a[3] = 7, a[4] = -8;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int size = 4;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0cout << solution(a, size);
}
```

2. Imagine you have a special keyboard with all keys in a single row. The layout of characters on a keyboard is denoted by a string S1 of length 26. S1 is indexed from 0 to 25. Initially, your finger is at index 0. To type a character, you have to move your finger to the index of the desired character. The time taken to move your finger from index i to index j is |j-i|, where || denotes absolute value.

Write a function solution(), that given a string S1 that describes the keyboard layout and a string S2,

returns an integer denoting the time taken to type string S2.

Examples:

S1 = abcdefghijklmnopqrstuvwxyz

S2 = cba

Input : S1 = abcdefghijklmnopqrstuvwxyz, S2 = cba \r\n**Output : 4**\r\n

```
#include <bits/stdc++.h>
using namespace std;
\xc2\xa0\xc2\xa0
int solution(string& s1, string& s2)
{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0map<char, int> dict;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0for (int i = 0; i < 26; i++) {
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0dict[s1[i]] = i;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int ans = 0;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0int prev = 0;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0for (int i = 0; i < s2.length(); i++) {
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0ans = ans + abs(dict[s2[i]] - prev);
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0prev = dict[s2[i]];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0return ans;
}
\xc2\xa0\xc2\xa0
int main()
{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0string s1 = "abcdefghijklmnopqrstuvwxyz";
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0string s2 = "cba";
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0cout << solution(s1, s2);
}
```

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'