B - Frog 2

Editorial (/contests/dp/tasks/dp_b/editorial)

/

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), the height of Stone i is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to one of the following: Stone $i+1, i+2, \ldots, i+K$. Here, a cost of $|h_i - h_j|$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Constraints

- All values in input are integers.
- $2 \leq N \leq 10^5$
- $1 \leq K \leq 100$
- $1 \leq h_i \leq 10^4$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & K & & \\
h_1 & h_2 & \ldots & h_N
\end{array}
$$

Output

Print the minimum possible total cost incurred.

2024-06-26 (Wed)
12:22:51 +05:30

Sample Input 1 Copy

```
5 3
10 30 40 50 20
```

Sample Output 1 Copy

```
30
```

If we follow the path $1 \rightarrow 2 \rightarrow 5$, the total cost incurred would be $|10 - 30| + |30 - 20| = 30$.

Sample Input 2 Copy

```
3 1
10 20 10
```

Sample Output 2 Copy

```
20
```

If we follow the path $1 \rightarrow 2 \rightarrow 3$, the total cost incurred would be $|10 - 20| + |20 - 10| = 20$.

Sample Input 3 Copy

```
2 100
10 10
```

Sample Output 3 Copy

```
0
```

If we follow the path $1 \rightarrow 2$, the total cost incurred would be $|10 - 10| = 0$.

Sample Input 4 Copy

```
10 4
40 10 20 70 80 10 20 70 80 60
```

Sample Output 4

Copy

```
40
```

Copy

If we follow the path $1 \rightarrow 4 \rightarrow 8 \rightarrow 10$, the total cost incurred would be $|40 - 70| + |70 - 70| + |70 - 60| = 40$.

Language C++ 20 (gcc 12.2)

Source Code

Open File

Customize

Toggle Editor

Auto Height

```
int dp[100100];
int rec(int level){
  if(level==n-1){
    return 0;
  }
  int ans=1e9;
  if(dp[level]!=-1){
    return dp[level];
  }
  for(int i=1;i<=k;i++){
    if(level+i<n){
      ans=min(ans,rec(level+i)+abs(arr[level]-arr[level+i]));
    }
  }
  return dp[level]=ans;
}
int main(){
  cin>>n>>k;
  memset(dp,-1,sizeof(dp));
  for(int i=0;i<n;i++){
    cin>>arr[i];
  }
  cout<<rec(0)<<endl;
  return 0;
}
```

* at most 512 KiB

* Your source code will be saved as Main.*extension*.

Submit

2024-06-26 (Wed)
12:22:51 +05:30