

Link to the P1 [https://atcoder.jp/contests/dp/tasks/dp\\_a](https://atcoder.jp/contests/dp/tasks/dp_a)

Understanding :-> You are standing at index 1 of the array. You have to reach the last index N of the array in minimum cost ; you can only make 1 step or 2 step jumps. If you are jumping from index i to index j cost will be  $\rightarrow \text{abs}(b[i] - b[j])$

General Way :->

-> First we declare empty  $\text{dp}[N]$

-> We assume  $\text{dp}[i]$  is the best answer to the question if size of the array is "i"

-> Always try to calculate the  $\text{dp}[1]$  ;  $\text{dp}[2]$  ;  $\text{dp}[3]$  ;  $\text{dp}[4]$  by yourself ; this will make you realise the pattern and you will be able to make the formula yourself

-> Then we try to create the formula.

-> We put the formula in the for loop and calculate  $\text{dp}[1]$  ,  $\text{dp}[2]$  ,  $\text{dp}[3]$  ,  $\text{dp}[4]$  .....  $\text{dp}[n]$

-> Generally  $\text{dp}[n]$  will be the final answer

[10 500 20 80]

$\text{Dp}[1]$  = minimum cost to reach the index1. = 0

$\text{Dp}[2]$  = minimum cost to reach the index2. = 490

$\text{Dp}[3]$  = best cost to reach index3 from index 1. = 10

$\text{Dp}[4]$  =  $\text{abs}(b[3]-b[4]) + \text{dp}[3]$  ) o1 (3 $\rightarrow$ 4) -> direct jump  
=  $\text{abs}(b[2]-b[4]) + \text{dp}[2]$  o2(2 $\rightarrow$ 4) -> direct jump

Answer will be the minimum of these two.

$$Dp[4] = \min(\text{abs}(b[3]-b[4]) + dp[3], \text{abs}(b[2]-b[4]) + dp[2])$$

General formula->

$$Dp[i] = \min(\text{abs}(b[i-1]-b[i]) + dp[i-1], \text{abs}(b[i-2]-b[i]) + dp[i-2])$$

Using this formula calculate  $dp[1] \dots dp[2] \dots dp[3] \dots dp[n]$

->  $dp[n]$  = final answer to reach the index n in minimum cost

Pseudo Code : - <https://ideone.com/mNiVpL>

TC :-  $O(N)$

SC :-  $O(N)$  ( $dp[n]$ )

C++ -> <https://atcoder.jp/contests/dp/submissions/42276696>

Java -> <https://atcoder.jp/contests/dp/submissions/42290564>

Python->

<https://atcoder.jp/contests/dp/submissions/42290680>

P2 :->

C++ -> <https://atcoder.jp/contests/dp/submissions/42277272>

Jv-> <https://atcoder.jp/contests/dp/submissions/42294397>

Py->

-> [https://atcoder.jp/contests/dp/tasks/dp\\_b](https://atcoder.jp/contests/dp/tasks/dp_b)

Understanding of Q2 :-> Same as above ; but you can make a jump of maximum "K" size.

Dp[i] = (i-1-->i)

OR

(i-2-->i)

OR

(i-3-->i)

OR

(i-4-->i)

OR

.

.

.

(i-k--->i)

Pseudo Code-> <https://ideone.com/XGv9SK>

TC : -  $O(N \cdot K)$

SC:-  $O(N)$  (DP[])