Frog 1

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 Stone $i+1$ or Stone $i+2$. 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.

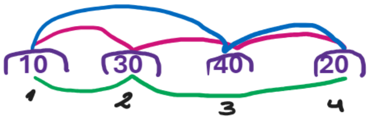
Sample Input 1 Copy

```
4
10 30 40 20
```

Sample Output 1 Copy

```
30
```

If we follow the path $1 \to 2 \to 4$, the total cost incurred would be $|10 - 30| + |30 - 20| = 30$.

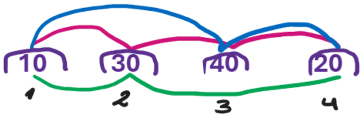


p(i) = min costo para alcanzar la piedra i

p(4) = | h(4) - h(3) | + |h(3) - h(2)| +|h(2) - h(1)|
p(4) = |20 - 40| + |40 - 30| + |30 - 10|
p(4) = 50

p(4) = | h(4) - h(3) | + |h(3) - h(1)|
p(4) = |20 - 40| + |40 - 10|
p(4) = 50

p(4) = | h(4) - h(2) | +|h(2) - h(1)|
p(4) = |20 - 30| + |30 - 10|
p(4) = 30



p(i) = min costo para alcanzar la piedra i

p(1) = 0

p(4) = | h(4) - h(3) | + |h(3) - h(2)| +|h(2) - h(1)|

p(4) = |20 - 40| + |40 - 30| + |30 - 10|

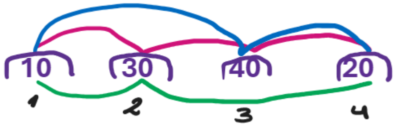
p(4) = 50

p(4) = | h(4) - h(3) | + p(3)

p(3) = |h(3) - h(2)| + p(2)

p(2) = |h(2) - h(1)| + p(1)

p(1) = 0



p(i) = min costo para alcanzar la piedra i

p(1) = 0

p(4) = | h(4) - h(3) | + |h(3) - h(1)|

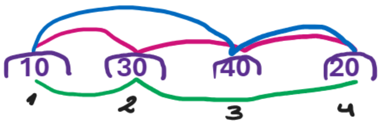
p(4) = |20 - 40| + |40 - 10|

p(4) = 50

p(4) = | h(4) - h(3) | + p(3)

p(3) = |h(3) - h(1)| + p(1)

p(1) = 0



p(i) = min costo para alcanzar la piedra i

p(1) = 0

p(4) = | h(4) - h(2) | +|h(2) - h(1)|

p(4) = |20 - 30| + |30 - 10|

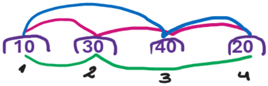
p(4) = 30

p(4) = | h(4) - h(2) | + p(2)

p(2) = |h(2) - h(1)| + p(1)

p(1) = 0

Frog 1



p(i) = min costo para alcanzar la piedra i

p(1) = 0

p(4) = | h(4) - h(2) | +|h(2) - h(1)|

p(4) = |20 - 30| + |30 - 10|

p(4) = 30

p(4) = | h(4) - h(3) | + p(3)

p(3) = |h(3) - h(2) | + p(2)

p(2) = |h(2) - h(1)| + p(1)

p(1) = 0

p(3) = |h(3) - h(1) | + p(1)

p(1) = 0

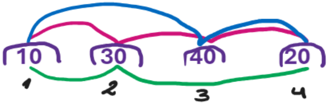
p(4) = | h(4) - h(2) | + p(2)

p(2) = |h(2) - h(1) | + p(1)

p(1) = 0

Overlap?

Frog 1: Overlap



$p(i)$ = min costo para alcanzar la piedra i

$p(1) = 0$

$p(4) = |h(4) - h(2)| + |h(2) - h(1)|$

$p(4) = |20 - 30| + |30 - 10|$

$p(4) = 30$

$p(4) = |h(4) - h(3)| + p(3)$

$p(3) = |h(3) - h(2)| + p(2)$

$p(2) = |h(2) - h(1)| + p(1)$

$p(1) = 0$

$p(3) = |h(3) - h(1)| + p(1)$

$p(1) = 0$

$p(4) = |h(4) - h(2)| + p(2)$

$p(2) = |h(2) - h(1)| + p(1)$

$p(1) = 0$

Frog 1: Formula

$p(i)$ = min costo para alcanzar la piedra i $\qquad p(1) = 0$

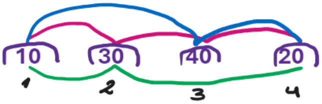
$p(4) = | h(4) - h(3) | + p(3)$

$$\boxed{p(i) = |h(i) - h(i-1)| + p(i-1)}$$

$p(4) = | h(4) - h(2) | + p(2)$

$$\boxed{p(i) = |h(i) - h(i-2)| + p(i-2)}$$

Frog 1: Formula



p(i) = min costo para alcanzar la piedra i

$p(1) = 0$

$p(4) = |\, h(4) - h(3) \,| + p(3)$

$p(i) = |h(i) - h(i-1)| + p(i-1)$

$p(4) = |\, h(4) - h(2) \,| + p(2)$

$p(i) = |h(i) - h(i-2)| + p(i-2)$

$$\mathbf{p(i) = \min(|h(i) - h(i-1)| + p(i-1),\ |h(i) - h(i-2)| + p(i-2))}$$

Frog 1

```java
private static int frog(int n) {
    if (n == 0) {
        return 0;
    }

    if (n == 1) {
        return Math.abs(h[0] - h[1]);
    }

    return Math.min(Math.abs(h[n] - h[n - 1]) + frog(n - 1),
        Math.abs(h[n] - h[n - 2]) + frog(n - 2));
}
```

Frog 1: Topdown DP

```java
private static int frog(int n) {
    if (n == 0) {
        return 0;
    }

    if (n == 1) {
        return Math.abs(h[0] - h[1]);
    }

    if (memo[n] != -1) {
        return memo[n];
    }

    memo[n] = Math.min(Math.abs(h[n] - h[n - 1]) + frog(n - 1),
        Math.abs(h[n] - h[n - 2]) + frog(n - 2));

    return memo[n];
}
```

Frog 1: Bottom-up DP

```java
private static int frog(int n) {
    int [] dp = new int[n];

    dp[1] = Math.abs(h[1] - h[0]);
    for (int i = 2; i < N; i++) {
        dp[i] = Math.min(Math.abs(h[i] - h[i - 1]) + dp[i - 1],
            Math.abs(h[i] - h[i - 2]) + dp[i - 2]);
    }

    return dp[n - 1];
}
```

Frog I

February 2024