Description | Editorial | Solutions (1.7K) | Submissions

714. Best Time to Buy and Sell Stock with Transaction Fee

Hint

Medium | 6.2K | 164

Companies

You are given an array `prices` where `prices[i]` is the price of a given stock on the i^{th} day, and an integer `fee` representing a transaction fee.

Find the maximum profit you can achieve. You may complete as many transactions as you like, but you need to pay the transaction fee for each transaction.

Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again).

Example 1:

```
Input: prices = [1,3,2,8,4,9], fee = 2
Output: 8
Explanation: The maximum profit can be achieved by:
- Buying at prices[0] = 1
- Selling at prices[3] = 8
- Buying at prices[4] = 4
- Selling at prices[5] = 9
The total profit is ((8 - 1) - 2) + ((9 - 4) - 2) = 8.
```

Example 2:

```
Input: prices = [1,3,7,5,10,3], fee = 3
Output: 6
```

Constraints:

- `1 <= prices.length <= 5 * 10^4`

Java | Auto

```java
class Solution {
    public int maxProfit(int[] prices, int fee) {
        int buy = Integer.MIN_VALUE;
        int sell = 0;

        for (int price : prices) {
            buy = Math.max(buy, sell - price);
            sell = Math.max(sell, buy + price - fee);
        }

        return sell;
    }
}
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

prices =
[1,3,2,8,4,9]

fee =
2

Output

Console | Run | Submit

Description | Editorial | Solutions (5.6K) | Submissions

191. Number of 1 Bits

Easy | 5.5K | 1.2K

Companies

Write a function that takes the binary representation of an unsigned integer and returns the number of '1' bits it has (also known as the Hamming weight).

Note:

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, the input will be given as a signed integer type. It should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
- In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 3**, the input represents the signed integer. `-3`.

Example 1:

```
Input: n = 00000000000000000000000000001011
Output: 3
Explanation: The input binary string
00000000000000000000000000001011 has a total of three '1' bits.
```

Example 2:

```
Input: n = 00000000000000000000000010000000
Output: 1
Explanation: The input binary string
00000000000000000000000010000000 has a total of one '1' bit.
```

Example 3:

Java | Auto

```java
public class Solution {
    public int hammingWeight(int n) {
    int count =0;

    for(int i=0; i<32;i++){
        if((n&1)==1){
            count++;
        }
        n>>=1;
    }
    return count;

    }
}
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

```
00000000000000000000000000001011
```

Output

```
3
```

Expected

```
3
```

Console | Run | Submit