Q17. Best time to buy and sell stock.

i/p → {7, 1, 5, 3, 6, 4}
o/p → 5 → maximum profit 6 − 1 = 5

Brute force
Take i = 0 and j = i+1 and then just find the maximum profit possible. j here will represent the selling Index & i represents the buying index and we don't have to care about the thing of first buying & then selling as j will always be ahead of i.

Optimal solution + Dry run
mini = INT_MAX;
profit = 0;

1) i = 0
prices [i] = 7
7 < INT_MAX ⇒ True & hence mini = 7

2) i = 1
prices [i] = 1
1 < 7 ⇒ True & hence mini = 1.

3) i = 2
prices [i] = 5
5 < 1 ⇒ False and hence
$\text{profit} = \max(0, \underset{\uparrow \atop \text{prices}[i]}{5} - \underset{\hookrightarrow \text{mini}}{1}) = \max(0, 4) = 4$

4) i = 3

prices [i] = 3
3 < 1 ⇒ False and hence
profit = max (4, 3 − 1) = max (4, 2) = 4

5) i = 4
prices [i] = 6
6 < 1 ⇒ False and hence
profit = max (4, 6 − 1) = max (4, 5) = 5

6) i = 5
prices [i] = 4
4 < 1 ⇒ False and hence
profit = max (5, 4 − 1) = max (5, 3) = 5

Hence we have now fully traversed the array and the profit comes out to be 5. Simply return profit.

Code

```
int max Profit (vector <int> & prices){
     // Setting the initial values
     int mini = INT_MAX;
     int profit = 0;
    // Traverse the whole array
    for (int i=0 ; i<prices.size(); i++){
          // Should buy if condition satisfied
          if (prices [i] <mini){
               mini = prices [i];
          }
          else { // Should sell
               profit = max (profit, prices [i] -
                                               mini)
```

```
            }
      }
      return profit;
}
```

Time complexity = O(n)
Space complexity = O(1)