Best time to buy and Sell Stocks infinite in C++

```
#include <iostream>
#include <vector>

using namespace std;

class
BestTimeToBuyAndSellStocksInfiniteTransactions {
public:
    int maxProfit(vector<int>& prices) {
        if (prices.empty()) return 0;

        int bd = 0; // Buy day
        int sd = 0; // Sell day
        int profit = 0;

        for (int i = 1; i < prices.size(); ++i) {
            if (prices[i] >= prices[i - 1]) {
                sd++;
            } else {
                profit += prices[sd] - prices[bd];
                bd = sd = i;
            }
        }

        profit += prices[sd] - prices[bd];
        return profit;
    }
};

int main() {
    BestTimeToBuyAndSellStocksInfiniteTransactions
solution;

    // Test case
    vector<int> prices = {11, 6, 7, 19, 4, 1, 6, 18, 4};
    int maxProfit = solution.maxProfit(prices);
    cout << "Max profit: " << maxProfit << endl; //
Output: 30

    return 0;
}
```

Dry Run

Input:

prices = {11, 6, 7, 19, 4, 1, 6, 18, 4}

Step-by-Step Execution:

- **Initialization**:
  - bd = 0, sd = 0, profit = 0.
- **Iterate Over Prices**:
  - **Day 1 (Price 6)**:
    - prices[1] < prices[0] → Calculate profit:
      - profit += prices[0] - prices[0] = 0 → No profit.
      - Update bd = 1, sd = 1.
  - **Day 2 (Price 7)**:
    - prices[2] >= prices[1] → sd = 2.
  - **Day 3 (Price 19)**:
    - prices[3] >= prices[2] → sd = 3.
  - **Day 4 (Price 4)**:
    - prices[4] < prices[3] → Calculate profit:
      - profit += prices[3] - prices[1] = 19 - 6 = 13.
      - Update bd = 4, sd = 4.
  - **Day 5 (Price 1)**:
    - prices[5] < prices[4] → No profit.
      - Update bd = 5, sd = 5.
  - **Day 6 (Price 6)**:
    - prices[6] >= prices[5] → sd = 6.
  - **Day 7 (Price 18)**:
    - prices[7] >= prices[6] → sd = 7.
  - **Day 8 (Price 4)**:
    - prices[8] < prices[7] → Calculate profit:
      - profit += prices[7] - prices[5] = 18 - 1 = 17.
      - Update bd = 8, sd = 8.
- **After Loop**:
  - Add remaining profit:
    - profit += prices[8] - prices[8] = 0.

Final Profit:

	• profit = 13 + 17 = 30.
Output:- Max profit: 30	