BEST TIME TO BUY AND SELL STOCK — Brute Force and Suffix Array Methods

Example Array:

prices = [7, 1, 5, 3, 6, 4]

Max profit = 5 (Buy at 1, Sell at 6)

PART 1 — BRUTE FORCE

Time: O(n^2)

Space: O(1)

Dry Run:

7-1=-6, 7-5=-2, 7-3=-4, 7-6=-1, 7-4=-3

1-5=4, 1-3=2, 1-6=5, 1-4=3

5-3=-2, 5-6=1, 5-4=-1

3-6=3, 3-4=1

6-4=-2

Max profit = 5

JAVA (Brute Force):

```
public static int maxProfitBrute(int[] prices){
int maxProfit=0;
for(int i=0;i<prices.length;i++){
for(int j=i+1;j<prices.length;j++){
maxProfit=Math.max(maxProfit,prices[j]-prices[i]);
}
}
return maxProfit;
}
```

C++ (Brute Force):

```
int maxProfitBrute(vector<int>& prices){
```

```
int maxProfit=0;
for(int i=0;i<prices.size();i++){
for(int j=i+1;j<prices.size();j++){
maxProfit = max(maxProfit, prices[j]-prices[i]);
}
}
return maxProfit;
}
```

Python (Brute Force):

```
def maxProfit_brute(prices):
maxProfit=0
for i in range(len(prices)):
for j in range(i+1,len(prices)):
maxProfit=max(maxProfit,prices[j]-prices[i])
return maxProfit
```

PART 2 — SUFFIX ARRAY METHOD

Time: O(n)

Space: O(n)

Suffix Max Array:

```
prices: 7 1 5 3 6 4
suffix: 7 6 6 6 6 4
```

JAVA (Suffix Array):

```
public static int maxProfitSuffix(int[] prices){
int n=prices.length;
int[] suffix=new int[n];
suffix[n-1]=prices[n-1];
for(int i=n-2;i>=0;i--)
suffix[i]=Math.max(prices[i],suffix[i+1]);

int maxProfit=0;

for(int i=0;i<n;i++)

maxProfit=Math.max(maxProfit,suffix[i]-prices[i]);

return maxProfit;

}
```

C++ (Suffix Array):

```
int maxProfitSuffix(vector<int>& prices){

int n=prices.size();

vector<int> suffix(n);

suffix[n-1]=prices[n-1];

for(int i=n-2;i>=0;i--)

suffix[i]=max(prices[i],suffix[i+1]);

int maxProfit=0;

for(int i=0;i<n;i++)

maxProfit=max(maxProfit,suffix[i]-prices[i]);

return maxProfit;

}
```

Python (Suffix Array):

```
def maxProfit_suffix(prices):

n=len(prices)

suffix=[0]*n

suffix[-1]=prices[-1]

for i in range(n-2,-1,-1):

suffix[i]=max(prices[i],suffix[i+1])

maxProfit=0

for i in range(n):

maxProfit=max(maxProfit,suffix[i]-prices[i])
```

```
return maxProfit
```