

# oppgave1

## Øving 1: Algoritmer og Datastrukturer

### Innhold

- [Oppgavebeskrivelse](#)
  - [Algoritme](#)
  - [Tidskompleksitet](#)
  - [Tidsmålinger](#)
  - [Konklusjon](#)
- 

### Oppgavebeskrivelse

Oppgaven gir endring i kurs på en aksje fra dag til dag. Algoritmen skal finne hvilken dag du burde kjøpe og selge for å få best fortjeneste.

---

### Algoritme

For hver dag:

1. Adderer endring i pris (input) til en variabel som lagrer den samlede prisen
2. Om prisen er det laveste den har vært er dagen en kandidat for å kjøpe, dette er  $\text{globalMin}$ ,  $\text{localMax} = \text{pris}$
3. Den høyeste prisen etter  $\text{globalMin}$  blir oppdatert settes som salgsdagen om  $\text{bestProfit} = \text{localMax} - \text{globalMin}$  er høyest

```
#include <iostream>
#include <chrono>

using namespace std;

int main()
{
    int n;
    cin >> n;

    auto start = chrono::high_resolution_clock::now();

    int bestProfit = 0;
```

```

int globalMin = 0;
int minDay = 0;
int buyAtDay = 0;
int sellAtDay = 0;
int price = 0;
for (int i = 0; i < n; i++)
{
    int priceChange;
    cin >> priceChange;

    price += priceChange;

    if (price < globalMin)
    {
        globalMin = price;
        minDay = i;
    }

    if (price - globalMin > bestProfit)
    {
        bestProfit = price - globalMin;
        buyAtDay = minDay;
        sellAtDay = i;
    }

    bestProfit = max(bestProfit, price - globalMin);
}

auto end = chrono::high_resolution_clock::now();

auto duration = chrono::duration_cast<chrono::microseconds>(end - start);

cout << "You make the best profit by buying at day " << buyAtDay << " and
selling at day " << sellAtDay << ".\nYour profit is then " << bestProfit << endl;

cout << "Time taken with n=" << n << " is " << duration.count() << "
microseconds." << endl;

return 0;
}

```

---

## Tidskompleksitet

Den verste og beste asymptotiske effektiviteten er  $n$ , altså har algoritmen en tidskompleksitet på  $(\Theta(n))$  siden den har én for-loop som kjører over alle dagene.

---

## Tidsmålinger

Input er generert med `generateInput.cpp` og skrevet til `input.txt`. Målingene er gjort med `chrono`.

( N )	KJØRETID (MIKROSEKUNDER)
10000	6783
100000	74655
1000000	619350

---

## Konklusjon

Etter å ha utført en praktisk tidsmåling med `chrono` ser vi at målingene stemmer overens med den teoretiske tidskompleksiteten altså  $O(n)$  og  $\Theta(n)$ .

---