

Rapport.Arbeidskrav1

Innhold:

- 1.Algoritmen
2. Kompleksiteten for tidsforbruket
3. Tidsmåling

Algoritmen

```
public static int[] get_profit(int[] values) {  
    int ind_min = 0, ind_max = 0, profit_main = 0;  
    for (int i = 0; i < values.length; i++) {  
        int profit_temp = 0;  
        for (int j = i + 1; j < values.length; j++) {  
            profit_temp += values[j];  
            if (profit_temp > profit_main) {  
                profit_main = profit_temp;  
                ind_min = i;  
                ind_max = j;  
            }  
        }  
    }  
    return new int[]{profit_main, ind_min, ind_max};  
}
```

Algoritmen inneholder løkker som finner midlertidig profitt og hvis den er mye enn profitten for hele tid, bytter vi hoved profitten og setter minimal indeks (dag da vi kjøper) og maksimum indeks (dag da vi selger). Først vi må kjøpe, og selge etterpå. Det gjøres med hjelp av at den andre løkken starter fra neste dag etter vi kjøper. Algoritmen retur profitt, dag da er det beste å kjøpe og dag da er det beste å selge. Det gjøres for output

Output med datas fra læreboken:

```
Best to buy: 3  
Best day to sale: 7  
Profit: 5  
Numbers: 9 | Millisekund pr. runde:6.461509866931511E-5
```

Det også ble lagt til mulighet for å se for hvilken tid ble algoritmen kjørt, og hvor mange dager er i array.

Kompleksiteten

Algoritmen inneholder en dobbeltløkke. Den indre går fra et bestemt tall til n, og er dermed $\Theta(n)$. Den ytre løkka går fra 0 til n, så dobbeltløkken skal blir $\Theta(n*n) = \Theta(n^2)$.

Tidsmåling

For å måle tiden laget jeg 3 tester. Tilfeldige matriser med lengder 100, 1000 og 10000. Hver array påfylles random med 5 unike verdier (Intervall fra -2 til 2). Resultater av testene kan du se under:

```
Number: 100 | Millisekund pr. runde:0.0034364497350497254  
Number: 1000 | Millisekund pr. runde:0.33987813134732564  
Number: 10000 | Millisekund pr. runde:35.714285714285715
```

Du kan se at hvis jeg øker antall av elementer i 10 ganger, tiden øker i 100 ganger. Det si meg at kompleksitet programmet mitt ble funnet helt riktig.