

Øving 1

Oppgave 1-1

```
public static String findGreatestInterest(int[] listOfValueShareDifference){

    int interest;
    int maxInterest=0;
    int buyDay=0;
    int sellDay=0;

    for (int i = 0; i < listOfValueShareDifference.length; i++) {
        interest=0;
        for (int j = i+1; j < listOfValueShareDifference.length; j++){
            interest+=listOfValueShareDifference[j];
            if(interest>maxInterest){
                maxInterest=interest;
                buyDay=i;
                sellDay=j;
            }
        }
    }
    return "Largest profit: " + maxInterest + "\n Buy day: " + (buyDay+1) + "\n Sell day: " + (sellDay+1);
}
```

Bilde 1: Bilde av algoritmen som vi utviklet

Oppgave 1-2

I algoritmen finnes det to for-løkker, der hver av dem skjer n-ganger, og den ene er inn i den andre. Den ytre for-løkken kan skrives som n, og den indre som C*n. Konstantene og de andre beregningene blir sett bort fra med tanke på at n^2 blir et veldig stort tall. Dette gjelder også for den konstante faktoren C. Kompleksiteten er derfor $O(n^2)$.

Oppgave 1-3

Dersom man observerer tabellen nedenfor man kan legge merke til at tidsmålingen stemmer overens med den asymptotiske analysen gjennomført i oppgave 1-2. Et eksempel på dette er endringen fra 410 ms til 1691 ms, altså en tilnærmet firedobling av kjøretid ved todobling av datamengde. Denne firedoblingen er tydeligere desto større tall det er snakk om (ifølge våre beregninger)

Lengde på tallrekke/n (datamengde)	Algoritmens kjøretid i ms
10000	30
20000	107
40000	410

80000	1691
160000	6847
320000	27550