

Morten Stavik Eggen
Vilhelm

Den første formelen kjører med n tid.

```
//Bruker denne for å oversikt på kursen, ble lettere å tenke da.
for (size_t i = 0; i < arrSize; i++)
{
    if (i == 0)
    {
        kurs[0] = 100;
    }
    else if (i != 0)
    {
        kurs[i] = kurs[i - 1] + (kurs[i - 1] * kursforandring[i] / 100);
    }
}
```

Kjører til størrelsen på arrayet.

Denne neste kjører med $O(n^2)$ tid fordi vi kjører en for loop i en for loop, så antall kjøringar blir i utgangspunktet $i \cdot i$ ganger.

```
//Sammenligner lave punkter mot høye punkter, setter Top og Bottom til den største forskjellen vi finner.
//Kunne blitt gjort mer optimalt med å bare sjekke punktene hvor det snur. kanskje noe lagring av verdier som er verdt å sjekke.
for (int i = 0; i < arrSize; i++)
{
    if (kurs[i] < tmpBottom && !(i > arrSize - 1))
    {
        tmpBottom = kurs[i];
        tmpTop = 0;
        //sjekk mot de høyere kursene etter denne dagen
        for (int j = i + 1; j < arrSize; j++)
        {
            if (kurs[j] > kurs[i])
            {
                tmpTop = kurs[j];
                //sjekk nye top/bunn med beste top/bunn
                if (top/bottom < tmpTop/tmpBottom){
                    top=tmpTop;
                    bottom=tmpBottom;
                    kjøp = i;
                    salg = j;
                }
            }
        }
    }
}
```

Antall kursforandringer	Tid
10 000	7994000ns
50 000	120996000ns
100 000	520968000ns

Som vi ser blir det ca 100 ganger så lang tid når vi ganger arrayet med 10. som stemmer godt med $O(n^2)$.

Morten Stavik Eggen
Vilhelm

```
[Running] cd "c:\Users\Morten\Desktop\School\Sem  
Tid brukt: 0[ns]  
Kjøps dag: 3  
Salgs dag: 7  
Kurs Kjøps dag: 93.73  
Kurs Salgs dag: 98.4724  
  
[Done] exited with code=0 in 1.419 seconds
```

Her er utskrift med dataen fra tabellen som ble oppgitt i oppgaven, men kan ikke ta tiden på den.

Kunne kanskje kjørt runder til vi når en viss mengde sekunder og se på hvor mange runder vi fikk.