Algoritmer og Datastrukturer Oppgave 1

1-1.

Koden er i den andre filen.

1-2.

Den asymptotiske effektiviteten er n^2. Det er fordi algoritmen inneholder to for-løkker.

O-notasjon:

$$O\big(g(n)\big) = \{f(n) : 0 \leq f(n) \leq cg(n) \; for \; alle \; n \geq n_0\}$$

C og n_0 er positive konstanter.

Det er mulig å analysere tidsforbruket nøyaktig, men det er ikke nødvendig for å finne kompleksiteten for tidsforbruket. Tidsforbruket i denne algoritmen er på formen $T(n) = A * n^2 + B * n + C$. For å finne kompleksiteten, er ikke konstantene relevante.

Viser at $f(n) \, \epsilon \, \Theta(n^2)$.

$$g(n) = n^2, f(n) = A * n^2 + B * n + C$$

$$0 \leq A * n^2 + B * n + C \leq c * n^2$$

$$0 \leq A + \frac{B}{n} + \frac{C}{n^2} \leq c$$

Så lenge n er større enn n_0, kan vi sette n_0 til et tall som er lik eller større enn C og B. Da kan c alltid bli settet til et tall som er større enn $A + \frac{B}{n} + \frac{C}{n^2}$. Når $n => \infty$ holder ulikheten.

Den innerste for-løkken vil først kjøres n ganger, så $n - 1$ ganger, så $n - 2$ osv. Dette fører til at den egentlig kjøres $\frac{n(n+1)}{2}$ ganger, men jeg forenkler fordi konstantene i $T(n)$ har ingen ting å si så lenge $n => \infty$.

1-3.

n = 100

```
Buy at day 5, where the price is 9995 and sell at day, 79 where the price is 10063. Price difference: 68.
Time used to run program: 0.000034
```

n = 1 000

```
Buy at day 48, where the price is 9972 and sell at day, 660 where the price is 10263. Price difference: 291.
Time used to run program: 0.002281
```

n = 10 000

```
Buy at day 4182, where the price is 9875 and sell at day, 9620 where the price is 10711. Price difference: 836.
Time used to run program: 0.148775
```

n = 100 000

```
Buy at day 69, where the price is 9915 and sell at day, 17230 where the price is 12288. Price difference: 2373.
Time used to run program: 10.948281
```

Fra n = 100 til n = 1 000 tar det 67 ganger lenger tid.
Fra n = 1 000 til n = 10 000 tar det 65 ganger lenger tid.
Fra n = 10 000 til n = 100 000 tar det 74 ganger lenger tid.

Dette resultatet er nærmere en kvadratisk vekst enn en lineær vekst. Dette stemmer med analysen fra oppgave 1-2 der jeg fant ut at algoritmen er kvadratisk.