Algoritmer og datastrukturer

Øving 1

Brage Minge

1-1. Har laget en algoritme som starter med dag 1 og sammenligner mulig avkastning om den selger på dag 2, så på dag 3 osv. Om avkastningen er større enn tidligere største avkastning lagres det nye paret av dager. Til slutt returnerer algoritmer en ArrayList som inneholder disse to dagene. Se Algorithm.java for hele algoritmen. I filen Main.java kan man og se at algoritmen fungerer på tabellen gitt i boka. Her legges alle kursendringene inn manuelt og algoritmen printer ut [3,7], som sier at beste profitt får man ved å kjøpe på dag tre og selge på dag syv, noe som er helt korrekt. Se bilde 1 og 2.

```
public class Main {

    public static void main(String[] args) {
        List<List<Integer>> list = RandomNumberListGenerator.fillListWithNumbers( amountOfNumbers: 1500,

        List<List<Integer>> testListe = new ArrayList<>();
        for(int i = 0;i<9;i++){
            testListe.add(new ArrayList<>());
        }
        testListe.get(0).add(-1);
        testListe.get(1).add(3);
        testListe.get(2).add(-9);
        testListe.get(3).add(2);
        testListe.get(4).add(2);
        testListe.get(5).add(-1);
        testListe.get(6).add(2);
        testListe.get(7).add(-1);
        testListe.get(8).add(-5);
        System.out.println(Algorithm.findDaysWithHighestProfit(testListe));
```

Bilde 1

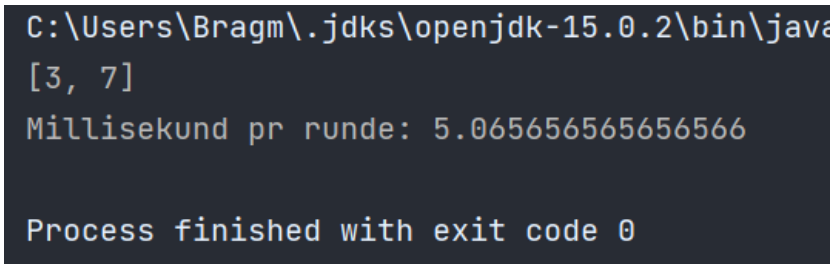
```
C:\Users\Bragm\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.1\lib\idea_rt.jar=57464:C:\Program Fi
[3, 7]
Millisekund pr runde: 5.025125628140704

Process finished with exit code 0
```

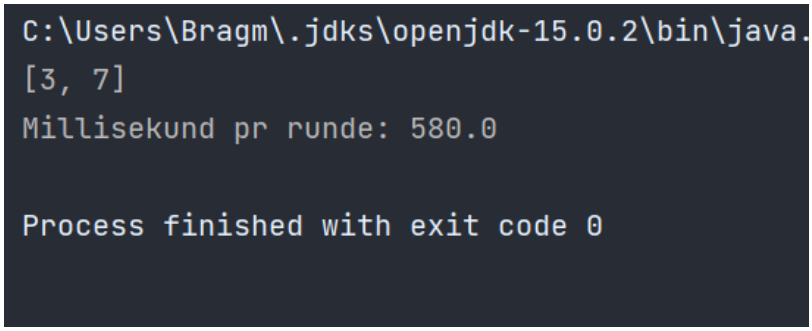
Bilde 2

1-2. Algoritmen min er av kompleksitet O(n^2). Dette fordi den inneholder en nestet for-loop, og kjøretiden vil da hundredobles ved en tidobling av n. Dette kommer og tydelig i neste oppgave.

1-3. Kompleksiteten kan tydelig sees på tidsmålingene. I bilde 3 kan man se at tiden per runde ved 1500 elementer er på omtrent 5 millisekunder. På bilde 4 er antall elementer i listen økt til 15.000 og tiden per runde er 580 millisekunder, altså rimelig nærme hundredobling, noe som er karakteristisk for en algoritme med kompleksitet O(n^2).



Bilde 3



Bilde 4