

Brage Minge

Algoritmer og datastrukturer

Øving 3

Andre innsending

Jeg har valgt å gjøre oppgave 2

Jeg fikk quicksort til å bli enda raskere som vist av tallene nedenfor. Bilde 1 er vanlig quicksort og bilde 2 er quicksort med counting sort. Vanlig quicksort snitter på omtrent 43/44 ms per runde på min maskin, og den forbedrede snitter på rundt 23 og 24 ms per runde.

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"  
Millisekund pr runde: 45.68181818181818  
  
Process finished with exit code 0
```

Bilde 1

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"  
Millisekund pr runde: 24.48780487804878  
  
Process finished with exit code 0
```

Bilde 2

Gitt et datasett der ingen av testene vil slå inn, bare kjøres gjennom, som f.eks. en tabell med tregangen, kan man se av bilde 4 at vanlig er mye raskere.

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-javaag  
Millisekund pr runde: 3.667033370003667E-4  
Millisekund pr runde vanlig: 1.1789599969441356E-4  
  
Process finished with exit code 0
```

Bilde 4

I klassen main.java, kan man se at den nye quicksort algoritmen klarer alle testene gitt i oppgaven, som å sortere en million tall og å sortere dem igjen på rimelig tid, uten feil.

Av bilde 5 og 6 kan man se hvilke tester som gjøres, og at ferdigsortert tabell sorteres igjen. På bilde 7 kan man se at de bekrefter at algoritmen virker som den skal.

```
10
11     int sumBefore = 0;
12     for (int i = 0; i < arr.length; i++) {
13         sumBefore += arr[i];
14     }
15
16     Algorithm.newQuicksort(arr, v: 0, h: arr.length-1);
17     int sumAfter = 0;
18     for (int i = 0; i < arr.length; i++) {
19         sumAfter += arr[i];
20     }
21     for (int i = 1; i < arr.length; i++) {
22         if (arr[i] < arr[i - 1] && arr[i] != arr[i - 1]) {
23             System.out.println("Wrong sorting");
24             break;
25         }
26     }
27     System.out.println(sumBefore == sumAfter);
28
```

Bilde 5

```
28
29
30     Algorithm.newQuicksort(arr, v: 0, h: arr.length-1);
31     sumAfter = 0;
32     for (int i = 0; i < arr.length; i++) {
33         sumAfter += arr[i];
34     }
35     for (int i = 1; i < arr.length; i++) {
36         if (arr[i] < arr[i - 1] && arr[i] != arr[i - 1]) {
37             System.out.println("Wrong sorting");
38             break;
39         }
40     }
41     System.out.println(sumBefore == sumAfter);
42
43 }
44
45
```

Bilde 6

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.  
true  
true  
  
Process finished with exit code 0
```

Bilde 7