

# DM507 Forår 2019 Projekt I

Antal beståede: 135

Antal genafleveringer: 31

## Generelle fejl:

| <i>Fejl ID</i> | <i>Beskrivelse</i>   |
|----------------|--|
| 1              | <p>Bruger 0-indekserede arrays, men ændrer ikke formlerne for børn og forældre-indekser (bogens formler er baserede på 1-indekserede arrays).</p> <p>Ved et lykketræf fungerer algoritmerne i bogen faktisk trods en sådan fejl (så man lægger ikke mærke til det under tests), men det er stadig teknisk set en forkert implementering af en heap.</p>  |
| 2              | <p>Bruger 0-indekserede arrays, men ændrer ikke værdien som l og r testes mod i IF-sætninger i heapify-metoden.</p> <p>Problemet er, at når man har udregnet left og right index, så skal man sørge for, at der rent faktisk ligger et element på left eller right før man ser på dem. I bogens pseudokode til Max-Heapify tjekker de fx om left findes på linje 3, ved at tjekke om "l &lt;= A.heap-size". Her bruger de "&lt;=", fordi deres arrays er 1-indekseret. Hvis man har ændret det til, at arrays er 0-indekseret, så skal man bruge "&lt;" hvis man tjekker op mod heap-sizen, ellers risikerer man at tjekke et element, som ikke eksisterer: hvis heapen fx kun består af 1 element, så er heap-sizen 1, og når man udregner left, så får man indeks 1 (som ikke er i brug), dvs. når man tjekker om "l &lt;= heap-size", så er det sandt, men så ender man med at kalde getKey() på en mulig null-pointer. Den måde heapify-metoden bliver brugt på i dette projekt, så bliver den kun kaldt i forbindelse med extractMin, dvs. den plads man kommer til at tjekke, hvor der ikke står et element, står der faktisk det element, som man netop har rykket op i roden, dvs. tilfældigvis virker koden altid (og man ender ikke med at kalde getKey() på en null-pointer).</p> |

|   |   |
|---|---|
|   | Dvs. det er teknisk set en forkert implementering af heapify, og man kunne forestille sig, at heapify kunne blive brugt i andre anledninger, og her vil den ikke altid mirakuløst virke.  |
| 3 | Bruger en ArrayList til at implementere heapen, selv om projektbeskrivelsen klart siger et almindeligt array (begrundelse herfor blev givet ved en forelæsning, som demonstrerede nogle mulige faldgruber ved brug af ArrayList).   |
| 4 | Bruger 1-indekserede arrays, men glemmer at man så i Java skal oprette et array som er én plads længere end antallet af elementer der skal være plads til (maxElms). Dette er fordi index 0 i array'et ikke bruges, men tæller med i arrayets længde under oprettelse.  |
| 5 | Afleveringen på papir og på Blackboard er ikke den samme.   |
| 6 | En afleveret fil er gemt som "utf8 med BOM". Filer skal gemmes som standard utf-8 (uden BOM), da Java kræver at der ingen BOM er, og ikke kan compilere "utf med BOM". Desværre vil Notepad under Windows bruge BOM når der gemmes i formatet utf8. Løsning: vælg kodning "ANSI" under "Save As" i Notepad.   |
| 7 | Man tæller antal elementer i heapen ved at løbe heapen igennem indtil et null element findes. Dette tager lineær tid. Man skal i stedet vedligeholde en tæller i PQHeap'en, som angiver antal elementer i heapen lige nu (variablen skal naturligvis opdateres under insert og extractMin). At læse denne tæller tager konstant tid, hvilket er nødvendigt for at vi alt i alt har logaritmisk køretid for insert og extractMin.                                      |
| 8 | Man følger ikke bogens pseudo-kode. Dette er krævet i opgaveteksten. Grunden er blandt andet at: 1) For bogens pseudo-kode har vi set argumenter, for at den er korrekt. Det har vi ikke for hjemmelavede pseudo-koder. 2) For bogens pseudo-kode har vi set argumenter for, at den kører i logaritmisk tid. Det har vi ikke for hjemmelavede pseudo-koder.   |
| 9 | Man bruger static modifier på variabler, som ikke bør være delt mellem alle instanser af PQHeap (f.eks. Heap-sizen og arrayet, hvor elementer opbevares i). Koden virker tilfældigvis, hvis man kun opretter ét PQHeap objekt nogensinde, men dette er ikke meningen. Se fx <a href="https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html">https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html</a> for forklaring bag static modifieren. |