Diskret Matematik og Algoritmer 2016

Ugeopgave 4

Rasmus F, Aiyu L & Frederik KM

Indhold

Del 1

Del 1.a

En node i en DH-liste skal indeholde en container til sin data, og 2 pointers som peger på det forrige og næste node i listen, eller null.

```
List::List (int z) : length (1) {
  head = new node(z)
  tail = head
}
```

Del 1.b

Vi går ud fra at S er sorteret i ikke-faldene orden

```
F(S, z) {
  i = S.head
  if (z < i.value) {S.insertBefore(i, z)}
  else
    while (i != null && z < (i.next).value())
      i = i.next
    S.insertAfter(i.prev, z)
}
```

Del 1.c

Alle linjerne i F kører i konstant tid, med undtagelse af while-løkken. Den kører fra head til - i værst tænklige tilfælde - tails; altså over de n elementer, og den gør det højest 1 gang. Altså er kørertiden $O(n)$.

Del 1.d

Hver gang et tal bliver indsat, skal den går over elementerne i listen en gang:

$$1 + 2 + ... + n - 1 + n = \frac{n^2 + n}{2}$$

Udtrykket oven over er $O(n^2)$

Del 2

Del 2.a

En måde man kunne gører dette på er at gennemløbe S, og for hvert k'te element opretter man en knude som man hægter på B. Kørertiden ville således være $O(n)$.

Del 2.b

Det ville tage $O(\sqrt{n})$, da dette er antallet af elementer i B.

Del 2.c

```
G(S, B, x)
 i = B.head
 while i != null && (i.value).value < x //Is S.value still smaller?
  i = i.next
 i = i.prev.value.next           // i is one past, decrement it and get next in S
 while i != null && i.value < x  //Is S.value still smaller?
  i = i.next
 i.prev.insertAfter(x)           //Decrement i and insert x after
```

Det er let at se at funktionen er $O(\sqrt{n})$, da den maksimalt kan gå over $\sqrt{n}$ elementer i B og $\sqrt{n}$ elementer i S; altså $2 \cdot \sqrt{n}$ som er $O(\sqrt{n})$.