Diskret Matematik og Algoritmer
Ugeopgave 4g

Mads Pontoppidan Haderup (xjr983), Ken Kjøller Andersen (zcj256)
og Christian Handest (lzp959)

4. oktober 2018

1 DEL 1

Lad S være en sorteret liste af tal repræsenteret ved hjælp af en dobbelthægtet liste (eng. double linked list). Målet med denne delopgave er at vedligeholde S når nye tal indsættes i S, således at S fortsat er sorteret.

1.1 DEL 1A

1. Beskriv hvilke felter (eng. attributter) en knude (eng. object/node) i listen S skal indeholde.

   Et knudeobjekt har 3 felter: Key, next og prev. Key er et heltal, next og prev er pegere som peger på henholdsvist næste og forrige objekt i S.

2. Lav pseudokode der givet et heltal z, opretter en liste med dette ene tal.

```
madeNode(z)
 node = new Node{prev = nil, key = z, next = nil}
 return node

makeList(z)
 S = new List{head = makeNode(z)}
 return S

// Test for z = 2
makeList(2)
```

1.2 DEL 1B

Lav pseudokode F(S,z), der tager en sorteret liste S, og et heltal z. Funktionen skal indsætte heltallet i S, sådan at S stadig er sorteret. Det skal ske på følgende måde: F gennemløber S fra starten indtil den finder det rigtige sted at indsætte heltallet z. Herefter indsættes z.

```
function makeNode(val)
  node = new Node{prev = NIL, key = val, next = NIL}
  return node

function SearchAndInsert(S, z)

    newNode = makeNode(z)

    temp1 = S.head
    temp2 = ''

    while newNode.key < temp1.key do

        if temp1.next == NIL then
            printfn "WE REACHED THE END"
            return -1
        else
            temp1 = temp1.next

    temp2 = temp1.next
    temp1.next = newNode
    newNode.prev = temp1
    newNode.next = temp2
    temp2.prev = newNode

    printfn "Success!
```

1.3 DEL 1C

Argumentér for at din funktion i værste fald bruger $\Theta(n)$ tid på indsættelse af et enkelt element, hvor n er antallet af tal i listen.

Funktionen har et while loop, som først stopper når det har fundet den korrekte placering for z i den sorterede liste S. Sagt med andre ord foregår der en lineær søgning, hvor while loopet ved med at iterere over nodernes keys, indtil z er større end den fundne key. I værste fald vil z skulle placeres bagerst i listen, hvilket svarer til O(n). Men

da z i alle andre tilfælde placeres forrest eller inde i listen, vil den gennemsnitlige placering blive betragtet i $\Theta(n)$ tid.

1.4 DEL 1D

Lad listen S være tom til at starte med. Der indsættes herefter n heltal. Et af gangen. Når det sidste heltal er indsat vil S være en sorteret liste bestående af de n indsatte tal. Argumentér for at vi på denne måde har sorteret n tal i O(n2) tid.
I ovenstående funktion insertAtEnd(n) indgår både et for- og nested while loop. Begge loops har en køretid på O(n) tid. Af den årsag vil funktionens samlede køretid blive O(n*n) = $O(n^2)$

1.5 DEL 1E

Hvilken sorteringsalgortime svarer forrige delopgave til?

Forrige algoritme svarer til insertion-sort. Da forrige algoritme sorterer som man ville sortere kort man får på hånden. ligeledes sorterer insertion sort. Dvs. man tager et kort, ser tallet, og sætter kortet ind på den plads, hvor tallet er højere end de forrige.

2 DEL 2

Lad S være en sorteret liste af n heltal som i ovenstående Del 1.

2.1 DEL 2A

Lad $n = k * k$ være således at $k = \sqrt{n}$ er et heltal. Listen S kan opdeles i k mindre sorterede lister, l_1, l_2 ... l_k, hvor hver af de k små lister, består af k elementer. Lav en dobbelthægtet liste B. Knuderne i listen B indeholder ikke tal som S, men pegere pa (ud over next/prev). Det i'te elements peger pa skal pege på listen li. Beskriv hvorledes S kan opdeles og B oprettes. Angiv ligeledes køretiden for dette.

En måde at gøre det på er at iterere over listen S for at finde n så kvadratroden af n kan findes. Hvis n er kendt i forvejen kan vi nøjes der med at finde kvadratroden af n. Herefter skal der oprettes en ny dobbelthægtet liste til at hvis første elements pa peger på det første element i listen S. Herefter kan der med en løkke der tager k skridt i S per iteration og tilføjer et element til B så elementets pa peger på elementer i listen S.

2.2 DEL 2B

Hvor lang tid, angivet i O-notation, tager det at gennemløbe listen B, hvis S indeholder n elementer?

Da køretiden for at gennemløbe listen S er O = (n), vil køretiden for at gennemløbe listen B som kun indeholder k elementer være O = ($\sqrt{n}$)

2.3 DEL 2C

Lav en funktion G der er givet S og B og et nyt heltal x. S indeholder $n = k * k$ elementer og B indeholder k elementer, som beskrevet ovenfor. G skal indsætte x i S ved at bruge B. Din funktion skal bruge tid $O(n\sqrt{n})$.

INSERT-SORTED-LIST(S, B, x)

```
1  b = B.nil.next
2  while b.pa ≠ B.nil  & b.pa ≠ x  & b.pa < x
3        b.pa = b.next
4        // Insert A[j] into the sorted sequence
5  i = b.pa
6  while i ≠ S.nil  & i.key ≤ x
7        i = i.next
8  x.next = S.i.next
9  S.i.next · prev = x
10 S.i.next = x
11 x.prev = S.i
```

2.4 DEL 2D

(frivillig – man behøves ikke at lave denne opgave) Vis at ovenstående tankegang kan bruges til at sortere n tal i $O(n\sqrt{n})$ tid.