Institut for Matematik og Datalogi
Syddansk Universitet, Odense

27. april 2025
RF

Opgaver Uge 18

DM507/DM578/DS814/SE4-DMAD

A: Løses i løbet af øvelsestimerne i uge 18

1. Eksamen juni 2010, opgave 1b:

   Spørgsmål b (5%):

   Angiv et Huffman-træ for en streng med følgende alfabet og tilhørende hyppigheder.

Tegn	a	b	c	d	e	f	g
Hyppighed	300	150	75	125	200	50	100

   □

2. Cormen et al., 4. udgave, øvelse 15.1-4 (side 425) [Cormen et al., 3. udgave: øvelse 16.1-4 (side 422)]:

   Du får opgivet en liste af forelæsninger, hver med en starttid s og en sluttid f, som angiver at forelæsningen dækker tidsintervallet $[s, f)$. Din opgave er at skemalægge alle forelæsningerne under brug af færrest mulig lokaler. Mere præcist: du har en masse lokaler til rådighed, som alle kan bruges til enhver af forelæsningerne, men du har lyst til at inddrage det færrest muligt antal forskellige lokaler i dit skema. Intet lokale kan på noget tidspunkt have mere end én forelæsning tildelt, naturligvis. Der behøver ikke være pause mellem to forelæsninger i samme lokale. Lav en "grådig" (eller i hvert fald simpel) algoritme, som laver en sådan skemalægning under brug af færrest mulige lokaler.

Hint: Lad a_t være antal aktiviteter som er i gang til tid t (dvs. $a_t = |\{i|s_i \leq t < f_i\}|$, hvor notationen stammer fra side 418 [Cormen et al., 3. udgave: side 415]). Lad t' være et tidspunkt t for hvilket a_t er maksimalt. Argumenter for at $a_{t'}$ er en nedre grænse for antal rum som skal bruges. Find så en simpel algoritme, som laver ét gennemløb fra venstre mod højre og foretager grådige/oplagte valg.

3. Cormen et al., 4. udgave, øvelse 15.1-3 (side 425) [Cormen et al., 3. udgave: øvelse 16.1-3 (side 422)]:

   I den grådig algoritme for bogens activity selection problem ser man på de aktiviteter, som ikke overlapper allerede valgte aktiviteter, og blandt dem vælger man (som det "grådige valg") den aktivitet, der slutter sidst. Andre naturlige forslag til grådigt valg kunne være:

   (a) Den aktivitet, der er kortest.

   (b) Den aktivitet, der overlapper færrest andre (tilbageværende) aktiviteter.

   (c) Den aktivitet, der starter først.

   For hvert af ovenstående andre forslag, find et modeksempel (dvs. en samling aktiviteter), som viser, at med dette forslag til grådigt valg vil den grådige algoritme ikke altid finde en optimal løsning (og den vil derfor ikke være en korrekt algoritme).

4. Cormen et al., 4. udgave, øvelse 14.1-2 (side 372) [Cormen et al., 3. udgave: øvelse 15.1-2 (side 370)]:

   Her er et forslag til grådig algoritme for guldkædeproblemet (kaldet rod cutting problemet i bogen): Sæt *værditætheden* af et stykke guldkæde af længde i til at være p_i/i, dvs. værdien per længde. For at finde en opklipning af en guldkæde af længde n, vælges den længde i, $1 \leq i \leq n$, som har størst værditæthed (et "grådige" valg). Denne klippes af, og man fortsætter på samme måde med resten af guldkæden (som nu har længde $n - i$).

   Find et modeksempel (dvs. en guldkædelængde n og en samling priser p_i), som viser, at denne algoritme ikke altid finder en optimal løsning (og derfor ikke vil være en korrekt algoritme).

5. Cormen et al., 4. udgave, øvelse 15.2-3 (side 430) [Cormen et al., 3. udgave: øvelse 16.2-3 (side 427)].

Bogen beskriver en grådig algoritme, som løser "fractional" versionen af Knapsack-problemet, men som fejler for 0-1 versionen af Knapsack-problemet.

Vi ser nu på følgende specielle typer input for 0-1 versionen af Knapsack-problemet: elementerne kan stilles i en rækkefølge, hvor der både gælder at vægten er stigende og at værdien faldende.

Vis, at bogens grådige algoritme løser 0-1 versionen af Knapsack-problemet for disse typer input.

6. (∗) Cormen et al., 4. udgave, øvelse 15.2-5 (side 431) [Cormen et al., 3. udgave: øvelse 16.2-5 (side 428)]:

   Givet en mængde $S = \{x_1, x_2, \ldots, x_n\}$ af tal på den reelle akse ønsker vi at finde det mindste antal lukkede intervaller af længde præcis én ($[2.45, 3.45]$ er et eksempel på et sådant interval), som tilsammen indeholder alle punkter i S. Find en grådig algoritme, som løser problemet. Argumenter for korrekthed og køretid.

   Hint: algoritmen er simpel, og ligner lidt den fra bogen, som løser activity selection problemet. For at bevise korrekthed af algoritmen, bevis følgende invariant: de hidtil valgte intervaller er en delmængde af en optimal dækning.

B: Løses hjemme inden øvelsestimerne i uge 19

Den sidste af nedenstående opgaver er opvarmning til projektet del III. Deltagere i DM578 og SE4-DMAD behøver derfor ikke lave denne opgave (men må gerne).

1. Eksamen januar 2008, opgave 1a:

   Spørgsmål a (8%): Betragt alfabetet med de syv tegn a, b, c, d, e, f, g. Nedenstående tabel viser, hvor tit hvert enkelt tegn optræder i en given tekst.

a	b	c	d	e	f	g
9	7	24	10	55	15	25

   Tegn et Huffman-træ, som repræsenterer Huffman-koderne for dette eksempel. □

2. Cormen et al., 4. udgave, øvelse 15.3-3 (side 439) [Cormen et al., 3. udgave: øvelse 16.3-3 (side 436)]:

   Vi ser her på input til Huffmans algoritme, hvor hyppighederne er givet ved Fibonacci-tal (Fibonacci-tallene er givet ved, at det næste tal er summen af de to forrige, samt at de to første begge er 1). For $n = 8$ er dette input således ud:

Tegn	`a`	`b`	`c`	`d`	`e`	`f`	`g`	`h`
Hyppighed	1	1	2	3	5	8	13	21

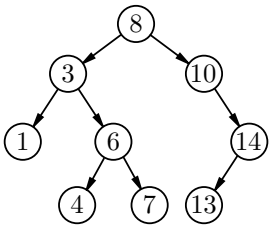
   Find et Huffman-træ for ovenstående input. Argumentér derefter for, hvordan Huffman-træet ser ud for sådanne input af vilkårlig størrelse n.

3. Cormen et al., 4. udgave, øvelse 15.3-7 (side 439) [Cormen et al., 3. udgave: øvelse 16.3-8 (side 436)]:

   Vi ser på input til Huffmans algoritme, hvor $n = 256$, og hvor hyppighederne ikke afviger mere end en faktor to fra hinanden. Mere præcist er den største hyppighed mindre end dobbelt så stor som den mindste hyppighed. Hvordan ser Huffman-træet ud for sådant input?

   Hint: Kan du sige noget om størrelsen af frekvenserne for undertræerne under de 128 første merge-skridt (som får de 256 originale træer (blot blade) til at blive til 128 træer)? Og igen under de 64 næste merge-skridt?

4. Opvarming til projektet del III: I jeres `DictBinTree` fra del II af projeket, tilføj en metode som udskriver en beskrivelse af stierne til alle knuder i træet. For følgende træ



   skal output være

```
Key 1: LL
Key 3: L
```

```
Key 4: LRL
Key 6: LR
Key 7: LRR
Key 8:
Key 10: R
Key 13: RRL
Key 14: RR
```

Hint: juster koden for inorder gennemløb passende.

Afprøv metoden på et træ bygget enten direkte i koden, eller via nogle kald til `insert`.

(NB: dette er ikke 100% hvad der skal ske i del III. Der er det f.eks. kun blade, som skal give output.)