Eksamensoppgaver

Vår 2020

Oppgave 2: Huffmantrær

I denne oppgaven skal du bruke et Huffmantre til å kode ordet «INTERCONNECTION».

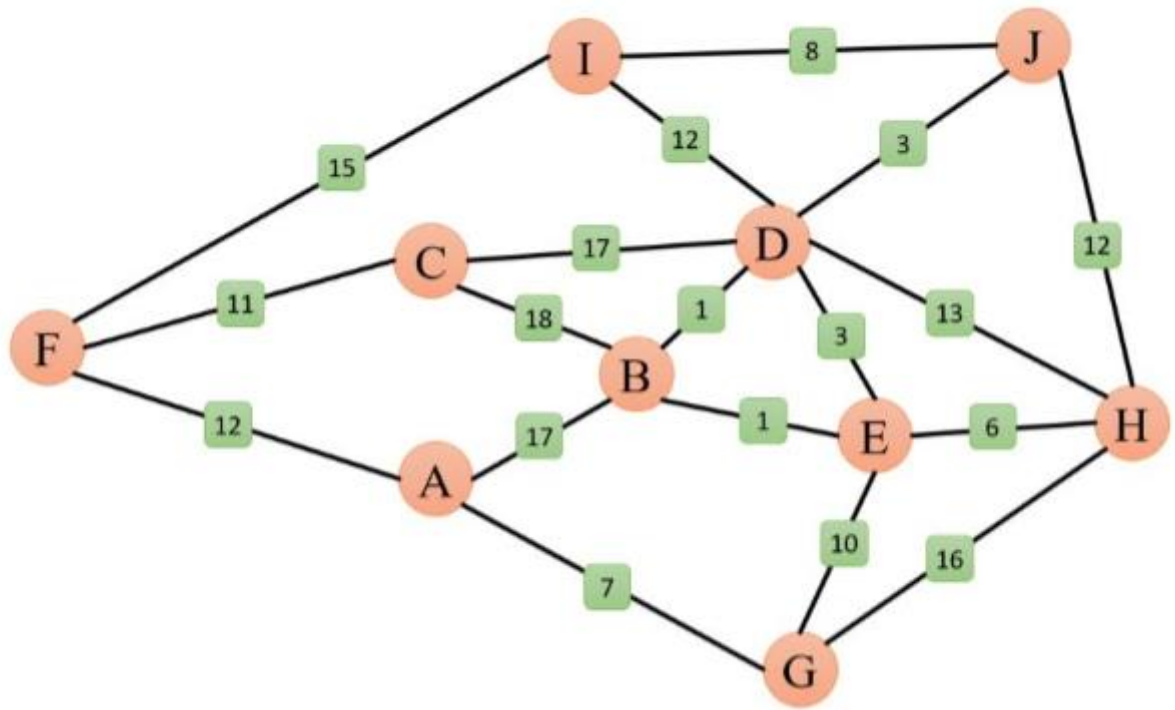
1. Lag Huffmantreet basert på ordet.
    1. Lag en tabell med frekvenser til hver bokstav,
    2. Lag en tegning av Huffmantreet, og
    3. Skriv opp Huffmankoden for hver bokstav.
2. Bruk så Huffmankodene til å komprimere ordet.
    1. Skriv opp den kodede binære meldingen.
    2. Hvor mange bit bruker den opprinnelige (ukodede) meldingen?
    3. Hvor mange bit bruker du i den kodede meldingen?

Vår 2019

Oppgave 4: Dijkstras algoritme

I denne oppgaven skal vi finne en vei i en graf ved hjelp av Dijkstras algoritme. I denne oppgavene er det viktig at du viser at du kan Dijkstras algoritme.

a. Beskriv med ord hva Dijkstras algoritme er.
    i. Hva bruker du Dijkstras algoritme til?
    ii. Hvordan fungerer Dijkstras algoritme?
    iii. Hvilken hjelpedatastruktur benyttes i Dijkstras algoritme?
b. Ta utgangspunkt i grafen i vedlegget. Lengden på hver kant står i firkantene.
    i. Lag en tegning som viser hvordan Dijkstras algoritme finner en vei fra F til H.
    ii. Hvilke noder passerer du og hva er lengden av veien?


I
J
F
C
D
B
E
H
A
G
8
15
12
3
12
17
11
18
1
3
13
12
1
17
6
12
10
16
7

Vår 2019

Oppgave 5: 2-3-4 trær (B-tre av orden 4)

I denne oppgaven skal du tegne forskjellige binære søketrær.

a. Start med et tomt binært søketre og legg inn tallene 15, 4, 11 (i den gitte rekkefølgen). Tegn treet.
b. Fortsett å legge inn tallene 9, 16, 29, 6 i treet. Tegn treet etter å ha lagt inn disse tallene.
c. Du skal nå legge inn de samme tallene i et 2-3-4 tre.
    i. Start med ett tomt tre og tegn treet etter å ha lagt inn 15, 4, 11.
    ii. Legg så til 9, 16, 29, 6, og tegn treet for hvert tall du legger til.
d. Hva er fordelen med å bruke et 2-3-4 tre i denne oppgaven?

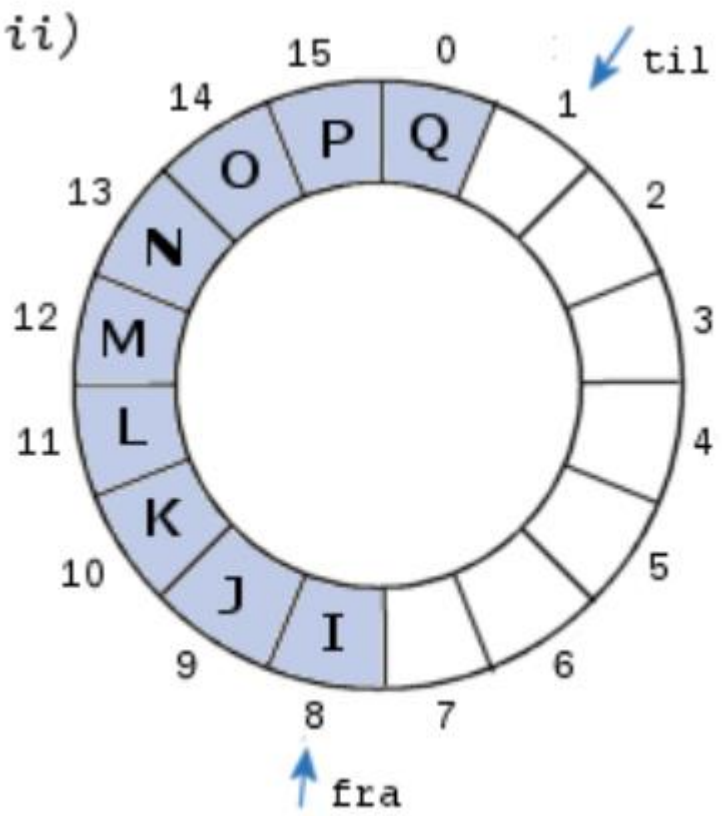
Høst 2016 2Bi

- i) En TabellKø oppfører seg som en vanlig kø. Dvs. leggInn() legger en verdi bakerst i køen og taUt() tar utden første verdien i køen. Hva blir utskriften i flg. programbit:

```
Kø<Character> kø = new TabellKø<>();
char[] c1 = "ABCDEFGHIJKLM".toCharArray(), c2 = "NOPQ".toCharArray();

for (char bokstav : c1) kø.leggInn(bokstav);
for (int i = 0; i < 8; i++) kø.taUt();
for (char bokstav : c2) kø.leggInn(bokstav);
System.out.println(kø);  // skriver ut køen
```

Høst 2016 2Bii

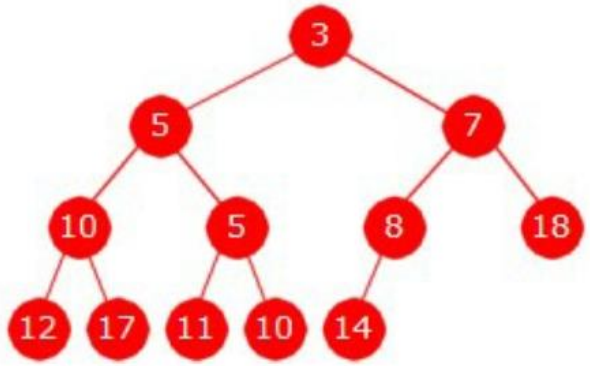


- Klassen TabellKø bruker internt en såkalt sirkulær tabell. Sefiguren til venstre. Der refererer indeks fra til den første i køen ogtil til den første ledige plassen (en bak den bakerste)..
- Legg inn R, S, T og U i den sirkulære køen til venstre og ta såut fem verdier. Tegn så køen (du behøver ikke bruke gråbakgrunn)! Hvor mange verdier har køen?.
- La tabellen hete a. Vis hvordan man generelt kan (så santtabellen ikke er full) finne antallet i en slik kø kun ved hjelp av verdiene til a.length, fra og til.

Vår 2019 3.2.2

- Forklar hvorfor Quicksort i gjennomsnittstilfellet har kompleksiteten O(n log(n))og i verste tilfellet har kompleksiteten O(n2)

Vår 2018 oppgave 5



- I denne oppgaven skal du jobbe med binær heap som datastruktur. Figuren i vedlagt bilde viser en binær heap(også kalt maksimumstre).
- a) Forklar hva som er spesielt med en binær minimumsheap: hvilke egenskaper må til for at det er en binærminimumsheap?
- b) Hva kan en binær minimumsheap brukes til?
- c) Gitt heap datastrukturen vedlagt bilde, legg til verdiene 6, 10, og 12. Tegn heapen for hvert tall du legger til.
- d) Forklar hvordan en tabell («array») kan brukes for å lagre en minimumsheap i minnet.
- e) Tegn tabellen som tilsvarer heap datastrukturen i figuren.

Vår 2018 oppgave 6

- Klassen LenketHashTabell bruker «lukket adressering med separat lenking». Den inneholderen tabell med nodereferanser der alle i utgangspunktet er null. Et objekt legges inn påobjektets tabellindeks (objektets hashverdi modulo tabellengden). Dvs. en node (med objektet) legges først iden (eventuelt tomme) lenkede nodelisten som hører til tabellindeksen.
- En samling navn (tegnstrenger) skal legges inn. Det er en jobb å regne ut hashverdier for hånd. Dette er derfor allerede gjort for noen lengder/dimensjoner:
- Setningen LenketHashTabell<String> hash = new LenketHashTabell<>(n); oppretter en instans av klassender den interne tabellen får dimensjon (lengde) lik n. Legg inn én og én verdi i den gitte rekkefølgen (dvs.Espen, Bo, Ali, osv). En node skal ha både verdi og hashverdi, men på en tegning holder det med verdi.

Vår 2018 oppgave 6

navn	Espen	Bo	Ali	Petter	Karl	Siri	Muhammad	Mari	August	Åse
hashverdi	80088	2357	65964	89125	69562	257197	7934	23763	65085	1983

- For enkelthets skyld velger vi å lage en hashtabell av lengde 10. Utfør følgende oppgaver med den lenkedehashtabellen:
- a) Regn ut indeksen for hver hashverdi i hashtabellen.
- b) Lag en tegning av datastrukturen når de fem første navnene er lagt inn
- c) Lag så en tegning som viser når alle navnene er lagt inn. Du trenger ikke å ta hensyn til tetthet (load factor)
- d) Hva betyr begrepet tetthet (load factor) i denne sammenheng, og hva må du gjøre om du skal ta hensyn tildette?
- e) Den lenkede hashtabellen vi har brukt har lengde 10: Er dette en god lengde? Hvorfor / hvorfor ikke?

Vår 2019 oppgave 3

- I denne oppgaven skal bruke flettesortering (merge sort) og kvikksortering (quicksort) til å sortereen tallrekke.
- a. Ta utgangspunkt i tallrekken 9, 8, 7, 6, 5, 4, 3, 2, 1. Lag en tegning som illustrererhvordan flettesortering (merge sort) sorterer tallene i stigende rekkefølge, og beskrivkort med tekst hva hovedprinsippet er.
- b. Ta utgangspunkt i samme tallrekke som i deloppgave a. Lag en tegning som illustrererhvordan kvikksortering (quicksort) sorterer tallene i stigende rekkefølge, og beskriv kortmed tekst hva hovedprinsippet er.
- c. Forklar hva begrepene verste tilfellet (worst case) og beste tilfellet (best case) betyr nårvi snakker om en algoritmes kompleksitet / effektivitet.

Høst 2020 kont

Oppgave 5: Minimumsheap

I denne oppgaven skal vi bruke en minimumsheap og se hvordan den kan brukes til sortering

1. Hva er en minimumsheap, og hvilke krav stilles for at det skal kunne kalles en minimumsheap?
2. Start med en tom minimumsheap. Legg tallene 5, 9, 3, 2, 6, 6, 1 og tegn heapen for hvert tall du legger inn.
3. Vi skal nå ta ut **tre** tall fra heapen. Ta ut ett og ett tall og tegn heapen for hvert tall du tar ut.
4. Forklar hvordan en minimumsheap kan brukes til sortering uten å bruke ekstra lagringsplass.