DMA 2021

– Ugeopgave 7 –

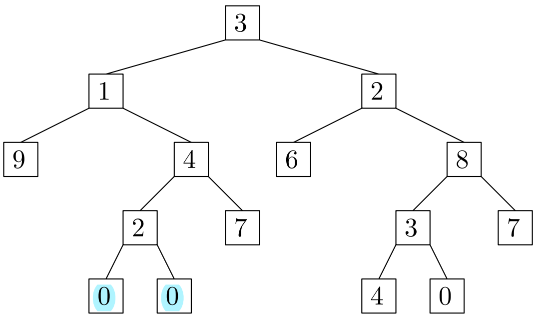
- Ugeopgaven skal skrives i Latex og afleveres som PDF.
- Ugeopgaven skal laves i grupper (2–3 personer).
- Navngiv opgaven som ugeopgave7navne.pdf hvor “navne” har noget med jeres navne at gøre.
- Alle spørgsmål skal forsøges besvaret.

Opgaven

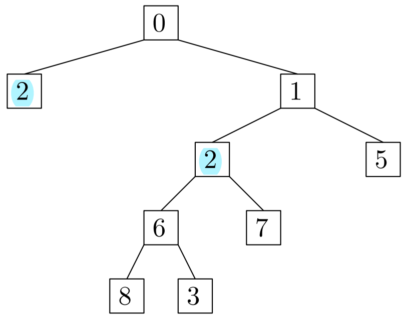
Del 1 **Vægtede binære træer:** Denne opgave omhandler rodfæstede binære træer. Hver knude har enten to eller ingen børn. Det venstre og højre barn af knuden x betegnes hhv. $x.left$ og $x.right$. Hvis knuden x ikke har nogen børn, har $x.left$ og $x.right$ den specielle værdi NIL. Til hver knude i træet er der knyttet en vægt; knuden x har vægten $x.weight$, og det oplyses at $x.weight \in \{0, 1, \ldots, n^2 - 1\}$, hvor n er antallet af knuder.

(a) Giv en effektiv algoritme $SameWeight(r)$ der givet rodknuden r til et vægtet binært træ returnerer 1 hvis to knuder i træet har samme vægt og 0 ellers. Køretiden skal være $O(n \log n)$ (men må naturligvis gerne være endnu mindre). Angiv køretiden af din algoritme og argumentér for at algoritmen er korrekt. **Bemærk:** Det er ikke et krav at vise pseudokode.

(b) To knuder i et træ er *søskende*, hvis de har samme forælder. Dvs. at for enhver knude x med to børn er $x.left$ og $x.right$ er søskende. Betragt et par $\{u, v\}$ af knuder som er søskende. Vi siger at knuden u er *let* hvis $u.weight \leq 2 \cdot v.weight$. Hvor mange lette knuder er der i hvert af træerne herunder?



(a)



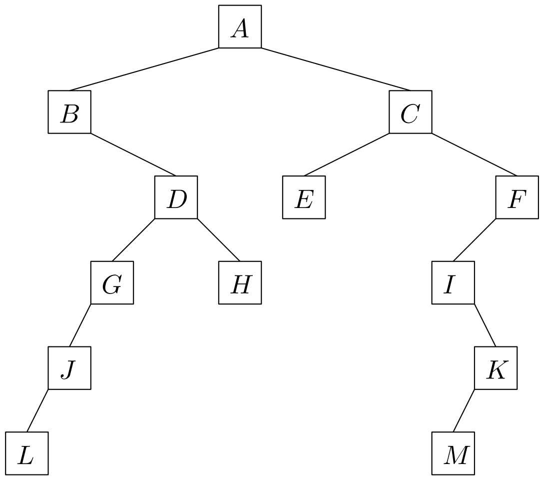
(b)

(c) Vis at der i et binært træ med n knuder er mindst $\frac{n-1}{2}$ og højst $n-1$ lette knuder, samt at disse grænser er tætte for alle ulige positive tal n, dvs. at der for alle ulige n findes et træ med n knuder og netop $\frac{n-1}{2}$ lette knuder, og et med netop $n-1$ lette knuder.

(d) Skriv pseudokode for en rekursiv algoritme $CountLightNodes(r)$, der givet rodknuden r til et vægtet binært træ returnerer antallet af lette knuder i træet. Angiv køretiden af din algoritme og argumentér for at algoritmen er korrekt.

Del 2 **Rekursion på træer:** Denne opgave omhandler rodfæstede binære træer. Enhver knude x har felterne $x.key$, $x.left$ og $x.right$, der betegner hhv. nøglen, det venstre barn og det højre barn. Felterne $x.left$ og $x.right$ kan være NIL, hvilket betyder at x ikke har noget venstre/højre barn.

(a) En knude i et binært træ er en *kædeknude* hvis knuden har netop ét barn. Angiv alle kædeknuder i nedenstående træ.



(b) Giv en algoritme $ChainNode(x)$, der givet en knude x returnerer TRUE hvis og kun hvis x er en kædeknude. Din algoritme skal køre i konstant tid. Skriv din algoritme i pseudokode.

(c) Giv en rekursiv algoritme $CountChainNodes(r)$, der givet rodknuden r returnerer antallet af kædeknuder i træet. Skriv din algoritme i pseudokode og analysér køretiden af din algoritme som funktion af n, hvor n er antallet af knuder i træet.

Del 3 **Hashtabeller med signaturer:** Denne opgave handler om at tælle antal forskellige ord i en tekst. Vi antager, at vi har en hashfunktion h, der ikke har nogen kollisioner på de ord, vi er interesserede i (vi kalder $h(x)$ en *signatur* for strengen x). Det vil sige at opgaven er at beregne antal forskellige værdier i en sekvens af signaturer $h(x_1), h(x_2), \ldots, h(x_n)$, hvor strengene $x_1, x_2, \ldots, x_n$ er ordene i teksten, som vi modtager ét af gangen.

(a) Antag at $h(x)$ er en w-bit værdi, og h har egenskaber der ligner en fuldt tilfældig funktion. Hvor stor kan længden n af sekvensen være hvis antagelsen om at der ikke er nogen kollisioner skal holde? Et svar i store-O notation er tilstrækkeligt.

(b) Giv en algoritme, der finder antal forskellige hashværdier i tid $O(n)$ og plads $O(n)$. Din algoritme skal virke selv om tallet n *ikke* er kendt i forvejen.

(c) Giv en forbedret analyse af algoritmens pladsforbrug når antallet af forskellige strenge er $m \ll n$.