DMA 2016

–Ugeopgave 7 –

- Hele ugeopgaven skal besvares.
- Ugeopgaven skal afleveres mandag den 5. december klokken 23:59 på Absalon.
- Ugeopgaven skal laves i **grupper** af 3-4 personer. *Bemærk at opgaven bruger grupperne* ***"Afleveringsgrupper, Opgave 7"****. I skal tjekke at jeres gruppe er korrekt tilmeldt til dette gruppesæt.*
- Besvarelsen skal udarbejdes i LaTeX.
- Del 3 af ugeopgaven er valgfri - man behøver ikke lave denne opgave.
- I Del 2 skal I bevise sætninger og udfylde manglende dele af et bevis. I kan vælge kun at aflevere de manglende dele, eller I kan udfylde de manglende dele i LaTeX koden, som ligger sammen med ugeopgaven på Absalon, og aflevere de fulde sætninger og beviser.

Del 1 Vi er interesserede i at implementere disjunkte mængder vha. en tabel/array A, således at $A[i]$ er repræsentant for i. Eksempelvis er mængderne $\{\mathbf{1}, 0, 6\}$ $\{8, \mathbf{3}, 2, 7\}$ og $\{4, \mathbf{5}\}$ repræsenteret ved tabellen $A = [1, 1, 3, 3, 5, 5, 1, 3, 3]$. Her er 1 f.eks. repræsentant for mængden $\{1, 0, 6\}$ fordi $A[0] = 1$, $A[1] = 1$ og $A[6] = 1$. Bemærk at repræsentationen *ikke* er identisk med tabelrepræsentationen beskrevet under "Hurtig forening" i noterne side 2.

(1) Skriv pseudokode for Init(n), Union(i, j) og Find(i) for denne implementation af disjunkte mængder med tabeller, og angiv køretiden for hver algoritme. Init(n) tager et tal n som argument og initialiserer en tabel med n elementer, hver som eneste element i sin egen mængde. Union(i, j) tager to elementer i og j og hvis de ikke tilhører samme mængde, forenes mængderne som i og j tilhører. Find(i) returnerer repræsentanten af den mængde som i tilhører.

(2) Håndkør følgende sekvens af operationer med din implementation af Init, Union og Find og vis indholdet af tabellen efter hver operation:
Init(7), Union(3,4), Union(5,0), Union(4,5), Union(4,3), Union(0,1), Union(2,6), Union(0,4), Union(6,0)

(3) Diskuter hvorledes implementationen adskiller sig fra implementationen gennemgået under "Hurtig forening" i noterne. Hvordan adskiller jeres version af UNION sig fra UNION som beskrevet på side 2 i noterne?

Del 2 Vi skal i denne opgave vise at højden på et træ som er fremkommet ved UNION med *union by rank* heuristikken (CLRS s. 571) er højst logaritmen af antallet af knuder i træet. Vi gør dette med tre sætninger nedenfor. I skal bevise eller udfylde de manglende dele af beviserne for de tre sætninger.

Notation: Når t er et træ som er fremkommet ved UNION med *union by rank* heuristikken lader vi $\text{rank}(t)$ være rangen af roden i t, og vi lader $\text{size}(t)$ være antallet af knuder i t.

Sætning 1. *Lad t være et træ som er fremkommet ved* UNION *af to træer t_1 og t_2 med* union by rank *heuristikken. Hvis* $\text{rank}(t_1) \neq \text{rank}(t_2)$ *er* $\text{rank}(t) \leq \max(\text{rank}(t_1), \text{rank}(t_2))$. *Hvis* $\text{rank}(t_1) = \text{rank}(t_2)$ *er* $\text{rank}(t) \leq 1 + \text{rank}(t_1)$.

(1) Lav et bevis for sætningen. (Hint: Kig på tilfældende i pseudokoden for Link CLRS s. 571.)

Sætning 2. *Lad t være et træ som er fremkommet ved at bruge* UNION *med* union by rank *heuristikken. Da er* $\text{rank}(t) \leq \log_2 \text{size}(t)$.

(2) Udfyld de to manglende dele (markeret med **. . .**) af induktionsbeviset nedenfor.

Bevis. Vi viser udsagnet med induktion over antallet af knuder n. Lad $P(n)$ være udsagnet:

$P(n)$: træer med n elementer, som er fremkommet ved at bruge UNION med *union by rank heuristikken*, opfylder $\text{rank}(t) \leq \log_2 n$.

Vi vil vise med stærk induktion at $P(n)$ er sandt for $n \geq 1$.

Induktionsstart: Vi skal vise at $P(1)$ er sandt. Lad t være et træ med 1 element.

. . .

Derfor er $P(1)$ er sandt

Induktionstrin: Vi skal vise at hvis $P(i)$ er sandt for alle $i = 1, \ldots, n-1$, så er $P(n)$ også sand. Vi kan derfor antage at $P(i)$ er sandt for alle $i = 1, \ldots, n-1$.

Lad t være et træ med n elementer. Træet er fremkommet ved UNION af to træer t_1 og t_2 med henholdvis i og j elementer, $1 \leq i, j \leq n-1$ og $i + j = n$.

Hvis $\text{rank}(t_1) = \text{rank}(t_2)$ har vi fra sætningen ovenfor og induktionsantagelsen

$$\begin{aligned}\text{rank}(t) &\leq 1 + \text{rank}(t_1) = \log_2(2) + \text{rank}(t_1) \leq \log_2(2) + \min(\log_2 i, \log_2 j) \\ &= \min(\log_2 2i, \log_2 2j) \leq \log_2(i + j) = \log_2(k)\ .\end{aligned}$$

Hvis $\text{rank}(t_1) \neq \text{rank}(t_2)$ **. . .**

I begge tilfælde betyder dette at $P(n)$ er sand.

Konklusion: Ved hjælp af induktionsprincippet ser vi nu at $P(n)$ gælder for alle $n \geq 1$.

$\square$

Sætning 3. *Lad t være et træ som er fremkommet ved at bruge* UNION *med* union by rank *heuristikken. Da er højden af t mindre eller lig $\log_2 \text{size}(t)$.*

(3) Lav et bevis for sætningen. Beviset kan referere til Sætning 2 ovenfor. I kan antage (dvs., I behøver ikke at vise) at $\text{rank}(t)$ er en øvre grænse for højden af et træ t, se CLRS s. 569.

Del 3 (Valgfri - man behøver ikke lave denne opgave) Som beskrevet i noterne kan UNION implementeret med tabel/array som i Del 1 udvides til en version WEIGHTED-UNION som vedligeholder et array *sz* hvor størrelsen af hvert træ er gemt på rodens plads. Lav et bevis med brug af induktion for et resultat som Sætning 3 ovenfor, men med WEIGHTED-UNION.