DMA 2021

– Ugeopgave 8 –

- Hele ugeopgaven skal besvares.
- Ugeopgaven skal laves **individuelt**.
- Besvarelsen skal udarbejdes i LaTeX.

Amortiseret analyse: Denne opgave handler om en speciel datastruktur til disjunkte mængder, der i modsætning til bogens eksempel ikke kan forenes, men i stedet kan skilles ad. Metoden, der skal bruges til analysen, minder om bogens analyse af disjunkt forening.

Lad F være en skov som består af binære træer med n knuder totalt. Vi ønsker at lave en datastruktur som understøtter tre operationer:

- $Initialize(F)$: Læg træerne fra F ind i datastrukturen. Denne metode kan kun kaldes én gang, når datastrukturen skabes.
- $Remove(F, x, y)$: Fjern kanten fra x til y fra F, så det træ der indeholder denne kant bliver splittet i to binære træer.
- $SameTree(F, x, y)$: Hvis x og y er knuder i det samme træ i F returneres TRUE, ellers returneres FALSE.

1. Betragt en kant e, som forbinder knuderne x og y i et træ T i F. Hvis vi fjerner e, bliver træet T til to mindre træer T_x og T_y, hvor x er en knude i T_x og y er en knude i T_y. Antag at $Initialize(F)$ kører i tid $O(n)$ og at $Remove(F, x, y)$ har køretid $O(\min\{|T_x|, |T_y|\})$, hvor $|T_x|$ og $|T_y|$ er antallet af knuder i hhv. T_x og T_y. Lad initielt F være et binært træ med n knuder og $n - 1$ kanter. Vis at operationerne kan gives følgende tidsgrænser:

   - $Initialize(F)$ tager amortiseret tid $O(n \log n)$
   - $Remove(F, x, y)$ tager amortiseret tid $O(1)$

   Hint: Du kan bruge regnskabsmetoden: Læg ved initialiseringen $\lg n$ dollars på hver knude. Hver knude x betaler 1 dollar hver gang der bliver fjernet en kant fra det træ som indeholder x og hvor x er i det *mindste* af de to resulterende træer. Argumentér for at en knude x i et træ T_x altid har mindst $\lg |T_x|$ dollars.

2. Givet knuder x og y i to forskellige træer T_x og T_y i F, forklar hvordan man kan lave en funktion $SmallestTree(x, y)$ som returnerer x hvis $|T_x| \leq |T_y|$ og y hvis $|T_x| > |T_y|$. Funktionen skal køre i tid $O(\min\{|T_x|, |T_y|\})$. Lav en figur som illustrerer hvordan funktionen virker. Du behøver ikke (men må gerne) skrive pseudokode.

   Hint: Gennemløb de to træer T_x og T_y parallelt.

3. Skriv pseudokode til funktionerne $Remove(F, x, y)$ og $SameTree(F, x, y)$, så $SameTree$ tager konstant tid i værste fald og de amortiserede grænser ovenfor gælder. Du kan bruge funktionen $SmallestTree(x, y)$ som black box, dvs. uden at skrive den ned.

   Hint: I hver knude x kan vi gemme et ID, $x.treeID$, på det træ som indeholder x. Hvis der er t træer kan mængden $\{1, \ldots, t\}$ bruges som ID'er. Lav funktionen $Remove(F, x, y)$ så den kan opdatere disse ID'er når en kant slettes og har køretid $O(\min\{|T_x|, |T_y|\})$.