Institut for Matematik og Datalogi
Syddansk Universitet, Odense

2. maj 2024
RF

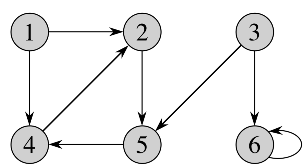
Opgaver Uge 19

DM507/DM578/DS814/SE4-DMAD

A: Løses i løbet af øvelsestimerne i uge 19

1. Cormen et al., 4. udgave, øvelse 20.2-1 (side 562) [Cormen et al., 3. udgave: øvelse 22.2-1 (side 601)]:

   Kør BFS(G, s) på nedenstående orienterede graf G, med knuden 3 som startknuden s. Angiv de resulterende d- og π-værdier.

2. Se på grafen G ovenfor som en uorienteret graf (dvs. se bort fra alle pilehoveder). Kør igen BFS(G, s) med knuden 3 som startknuden s. Angiv de resulterende d- og π-værdier.

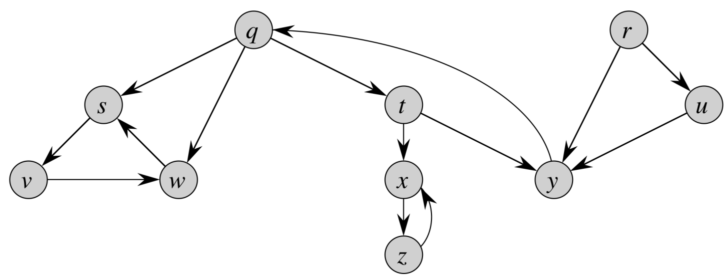
3. Cormen et al., 4. udgave, øvelse 20.2-3 (side 562) [Cormen et al., 3. udgave: øvelse 22.2-3 (side 602)]:

   Forklar, hvorfor den sidste linje i BFS(G, s) kan udelades uden at ændre algoritmens opførsel. [Dette viser, at farverne hvid og ikke-hvid er nok, hvilket kun kræver én bit at opbevare i knuder.] Hint: bruges forskellen grå/sort til at tage beslutninger i algoritmen?

   Ekstraopgave: forklar, hvorfor heller ikke denne bit er nødvendig i BFS. [Hint: se på d-værdierne i stedet. (Hvorfor kan π-værdierne ikke bruges?)]

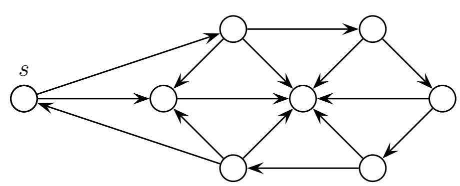
4. Cormen et al., 4. udgave, øvelse 20.3-2 (side 571) [Cormen et al., 3. udgave: øvelse 22.3-2 (side 610)]:

   Kør DFS(G) på nedenstående graf. Antag, at knuder er ordnet alfabetisk i **for**-loopet i DFS(G), og at nabolister er ordnet alfabetisk i **for**-loopet i DFS-VISIT(G, u). Angiv de resulterende d- og f-værdier for knuder og de resulterende kanttyper (tree, back, forward, cross) for kanter.



5. Eksamen juni 2010, opgave 2, spørgsmål a og b:

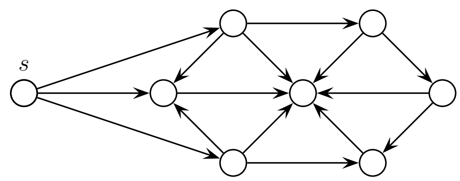
Spørgsmål a (6%): For alle knuder v i grafen G_1, angiv distanceværdien $v.d$ som tildeles ved bredde-først søgning (BFS) med start i knuden s.



Figur 1: Grafen G_1

Spørgsmål b (7%): For alle knuder v i grafen G_2, angiv starttiden (discovery time) $v.d$ og sluttiden (finishing time) $v.f$ som tildeles ved dybde-først søgning (DFS) med start i knuden s.

(For DFS afhænger det præcise resultat af ordningen af knuders nabolister. Du skal her antage at på figuren er alle knudes nabolister ordnet "med uret", startende fra "lodret opad".)



Figur 2: Grafen G_2

6. Cormen et al., 4. udgave, øvelse 20.3-4 (side 571) [Cormen et al., 3. udgave: øvelse 22.3-4 (side 611)]:

   Forklar, hvorfor linje "$u.color$ = BLACK" i DFS-VISIT(G, u) kan udelades uden at ændre algoritmens opførsel. [Dette viser, at farverne hvid og ikke-hvid er nok, hvilket kun kræver én bit at opbevare i knuder.] Hint: bruges forskellen grå/sort til at tage beslutninger i algoritmen?

   Ekstraopgave: hvordan kan man i DFS bruge d-værdierne i stedet for denne bit?

7. Cormen et al., 4. udgave, øvelse 20.3-9 (side 572) [Cormen et al., 3. udgave: øvelse 22.3-10 (side 612)]:

   Forklar, hvordan pseudo-koden for DFS kan udvides til at udskrive typen for alle kanter i inputgrafen G. Antag først, at G er orienteret. Gentag derefter under antagelse af, at G er uorienteret.

   Note: For denne udvidelse af DFS er det nødvendigt, at DFS bruger alle farverne hvid, grå og sort (modsat Cormen et al., 4. udgave øvelse 20.3-4 ovenfor).

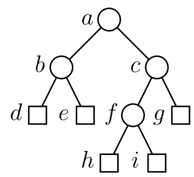
De næste to opgaver er repetition af tidligere stof.

8. Eksamen juni 2012, opgave 1:

Spørgsmål a (5%):

Angiv en farvning af knuderne i træet T_1 som gør det til et rød-sort træ. (Svar ved at skrive en liste af navnene på de sorte knuder og en liste med navnene på de røde knuder.)

T_1:

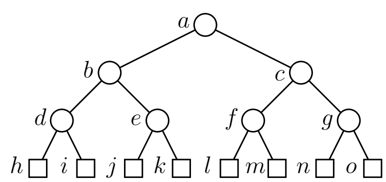


□

Spørgsmål b (5%):

Angiv *alle* farvninger af knuderne i træet T_2 som gør det til et rød-sort træ. (Svar ved for hver farvning at skrive en liste af navnene på de sorte knuder og en liste med navnene på de røde knuder.)

T_2:



□

9. Eksamen juni 2012, opgave 3:

Spørgsmål a (7%):

Angiv løsningen til følgende rekursionsligning.

$$T(n) = 8 \cdot T(n/4) + n^{1.5}$$

Spørgsmål b (8%):

Angiv for hver af følgende rekursionsligninger *om* de kan løses ved hjælp af master theorem (Theorem 4.1) i lærebogen. For hver ligning hvor svaret er positivt, angiv hvilken af de tre cases i master theorem som løser den. (Du behøver ikke angive selve løsningen.)

$$\begin{array}{rl} \text{i)} & T(n) = 14 \cdot T(n/13) + n \\ \text{ii)} & T(n) = 13 \cdot T(n/13) + n \log n \\ \text{iii)} & T(n) = 14 \cdot T(n/13) + n \log n \\ \text{iv)} & T(n) = 13 \cdot T(n/14) + n \end{array}$$

I spørgsmål b, angiv også løsningerne (selv om teksten siger det modsatte). Bemærk at mængden af rekursionsligninger, som *kan* løses med master theorem, er steget fra 3. til 4. udgave af lærebogen. Vi skal her (og til eksamen) bruge versionen fra 4. udgave (som også findes på slides).

B: Løses hjemme inden øvelsestimerne i uge 20

1. Cormen et al., 4. udgave, øvelse 20.4-3 (side 575) [Cormen et al., 3. udgave: øvelse 22.4-3 (side 615)]:

   Lav en algoritme, som kan afgøre, om en uorienteret graf indeholder en kreds (cycle). Algoritmen skal køre i tid $O(|V|)$, uafhængigt af værdien af $|E|$.

   Der må bruges, at hvis en uorienteret graf er acyklisk (ikke indeholder en kreds), så er $|E| \leq |V| - 1$ (dette følger af sætning B.2, punkt 5 og 6 (side 1170 [Cormen et al., 3. udgave: side 1174])).

2. ($*$) Cormen et al., 4. udgave, øvelse 20.2-7 (side 563) [Cormen et al., 3. udgave: øvelse 22.2-7 (side 602)]:

   En uorienteret graf $G = (V, E)$ kaldes *bipartite*, hvis knudemængden V kan deles i to delmængder A og B, således at alle kanter har det ene endepunkt i A og det andet i B.

   Lav en algoritme, som i tid $O(|V| + |E|)$ kan afgøre, om en given uorienteret graf er bipartite. Hvis svaret er ja, skal algoritmen også returnere en mulig opdeling i A og B.

   Hint: Udvid BFS passende.