Algoritmer & Datastrukturer
Aflevering 4

Anders Bjerg Pedersen, Hold 2

7. marts 2007

Excercise 1

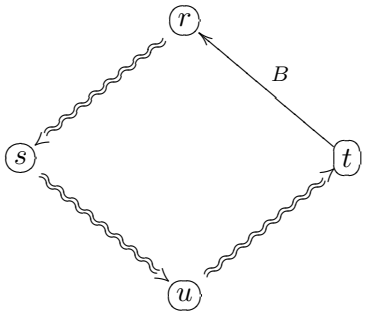
Vi kører den modificerede udgave af DFS, der også farver kanter efter konventionerne på side 546 i bogen, på en tilfældig valgt knude i G. Hvis der findes en knude u undervejs med $d[u] = f[u] - 1$, vælges denne som vores startknude r (vi har da fat i et blad), eller vi vælger en knude, som vil kunne nåes vha. en back edge (da denne så vil indgå i en simpel cykel). Altså kan vi anvende DFS til at finde vores startknude.

Excercise 2

Lad G' være den graf, der er fremkommet efter kørsel af algoritmen på den ikke-orienterede sammenhængende graf G. Vi vil vise, at der ikke kan findes en knude $u \in G'$ med enten $indegree(u) = 0$ eller $outdegree(u) = 0$, uden at den også opfylder kravene for en balanceret orienteret graf. Altså at G' er en balanceret orienteret graf.

Da G før vi kørte DFS(G) var en ikke-orienteret graf, vil DFS(G) resultere i en DFS-skov med kun ét træ. Dette følger af Theorem 22.10 og det faktum, at G i vores tilfælde er sammenhængende. Dette ene træ medfører nu, at fra vores startknude r vil der jfr. Theorem 22.9 gælde, at alle knuder pånær r selv vil være efterkommere til r.

- Antag, at der findes en knude $u \in G'$ med $indegree(u) = 0$. Der kan ikke eksistere en knude med denne egenskab, da enhver knude bortset fra rodknuden er efterkommer til rodknuden (jfr. ovenfor) og dermed har *indegree* mindst 1. Altså må der være tale om rodknuden. Denne rodknude kan ej heller have indgået i en simpel cykel i G, da vi under DFS(G) så ville have fået en back edge tilbage til den og dermed en *indegree* på mindst 1 (se eksempelvis Figur 1 herunder). Altså må rodknuden have *degree*=1, hvormed G' opfylder betingelserne for en balanceret orienteret graf.



Figur 1: Skitse af cykel i DFS

- Antag, at der findes en knude $u \in G'$ med $outdegree(u) = 0$. Vi deler op i to tilfælde:
  - $indegree(u) = 1$: Her er der tale om, at u er et blad, altså er kravene for en balanceret orienteret graf opfyldt.
  - $indegree(u) \geq 1$: Dette kan ikke lade sig gøre. Når vi besøger en knude med $degree \geq 1$ første gang, vil vi under kørslen af algoritmen fortsætte ud af en af de andre kanter (enten som en back edge eller en tree edge), hvormed vi får, at knuden vil få $outdegree \geq 1$, hvilket er i modstrid med vores antagelse.

Altså har vi vist, at algoritmen kører korrekt. Med hensyn til køretiden vil trin 1 skulle udføre en eller flere DFS'er, der muligvis stopper, inden de er kørt færdige. Trin 2 udfører ligeledes en DFS, der denne gang kører fuldt ud. Køretiden er derfor opad begrænset: $O(2(V + E)) = O(V + E)$.

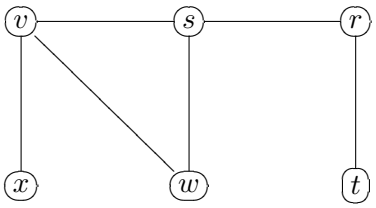
Excercise 3

Den nedre grænse for antallet af kanter E er $V - 1$ i bedste tilfælde i en sammenhængende graf som vores. Trin 1 vil i bedste tilfælde kunne køre i $\Omega(1)$, hvis vores startknude r findes i første hug. I trin 2 vil vi ligeledes i bedste tilfælde have, at $E = V - 1 \Rightarrow V = E + 1$. Køretiden for DFS er $O(V + E)$, hvilket i bedste tilfælde så bliver til $\Omega(V + E) = \Omega(E + 1 + E) = \Omega(2E) = \Omega(E)$. Grunden til, at vi kan se bort fra V's betydning er, at vi har at gøre med en sammenhængende graf. Med hensyn til en øvre grænse, så gælder der, at antallet af kanter maksimalt kan være $\binom{n}{2}$, der jo som bekendt vokser meget hurtigere end n selv. Dette vil betyde, at der for en graf med $|V| = n$ knuder vil gælde, at antallet af kanter vil dominere, hvilket vil betyde, at algoritmen asymptotisk kommer til at køre i $O(E)$ tid.

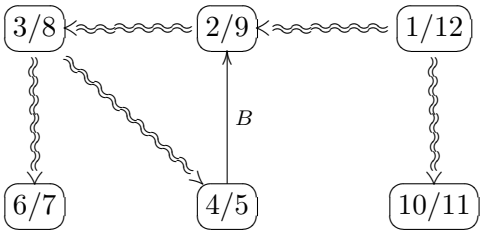
Alt i alt har vi så, at algoritmen kører i $\Theta(E)$ tid.

Excercise 4

Se på følgende eksempel, hvor vi udelader trin 1 i algoritmen og starter i knude r:



Figur 2: Ikke-orienteret graf, hvor vi starter DFS i knude r.



Figur 3: Efter DFS(r). Det gik galt...

Her er startknuden r netop valgt, så den ikke er enten et blad eller indgår i en simpel cykel. Det går galt, fordi $outdegree(r) = 2$ og $indegree(r) = 0$. Enhver anden knude ville have været med i en simpel cykel eller have været et blad, hvorfor algoritmen ville have virket.