OPEN BOEK EXAMEN
DATASTRUCTUREN EN GRAAFALGORITMEN
14/6/2019

Oefeningen (3,3,3,4,3,2,2)

1. GRAAF SEARCHING (3): Geef een voorbeeld van een graaf $G = (V, E)$ met $V = \{1, \ldots, n\}$ en $|E| = n(n+1)/2 - 1$ zodat BFS(G,1) en DFS(G) dezelfde output geven wanneer alle adjacency lists oplopend gesorteerd zijn. [Hint: start met n klein.]

2. GRAAF SEARCHING ($3 = 2 + 1$): Gegeven een graaf $G = (V, E)$ met SCCs $C_1, \ldots, C_k$. Laat $V(C_i) = \{u \in V | u \in C_i\}$, $E(C_i) = \{(u, v) \in E | u, v \in C_i\}$, $E(C_i, C_j) = \{(u, v) \in E | u \in C_i, v \in C_j\}$. Stel dat $G = (V, E')$ dezelfde k SCCs heeft. Wat is de kleinst mogelijke waarde van $|E'|$? Wijzigt je antwoord indien we eveneens eisen dat $E' \subseteq E$? Leg uit.

3. FLOW NETWERKEN (3): Laat f een flow zijn in $G = (V, E)$, f^* een maximale flow en $G_f(\Delta)$ het residual netwerk van G tov de flow f waarbij we alle edges met $c_f(u, v) \leq \Delta$ hebben verwijderd. Toon aan dat $|f^*| \leq |f| + \Delta|E|$ wanneer er geen pad is van s naar t in $G_f(\Delta)$. [Hint: Kies een bepaalde cut (S', T') en maak gebruik van het feit dat $|f^*| \leq c(S, T)$ voor elke cut (S, T).]

4. FLOW NETWERKEN (4=1.5+1.5+1): Gebruik de notatie van de voorgaande oefening. Stel dat alle edges een **gehele capaciteit hebben tussen** 1 **en** C. Beschouw volgende code:

   $f = 0$; $\Delta = 2^{\lfloor \log_2 C \rfloor}$
   while $\Delta \geq 1$ **do**
       while There exists a path p from s to t in $G_f(\Delta)$ **do**
           augment f with f_p
       end while
       $\Delta = \Delta/2$
   end while

   Beantwoord volgende vragen en leg uit (gebruik de eigenschap in de voorgaande oefening indien nodig):

   (a) Eindigt dit algoritme steeds en is f op het einde een maximale flow?
   (b) Hoe vaak voeren we elk van de while lussen maximaal uit?
   (c) Wat is de totale complexiteit van dit algoritme?

5. DISJOINT SET DATA STRUCTURES ($3 = 2 + 1$): Gegeven een graaf $G = (V, E)$. Wat wordt er getest door onderstaand algoritme? Geef voldoende uitleg. Gebruiken we best de linked-list of forest implementatie?

   for $u \in V$ **do**
       MAKESET(u);
   end for
   Select random $e' = (u', v') \in E$; $E' = E \setminus e'$.
   while $E' \neq \emptyset$ **do**
       for $e = (u, v) \in E'$ **do**
           if FINDSET(u) = FINDSET(v) **then**
               Return False;
           else
               if FINDSET(u) = FINDSET(u') **then**
                   UNION(FINDSET(v),FINDSET(v')); $E' = E' \setminus e$;
               end if
               if FINDSET(u) = FINDSET(v') **then**
                   UNION(FINDSET(v),FINDSET(u')); $E' = E' \setminus e$;

        end if
        if FINDSET(v) = FINDSET(u') **then**
            UNION(FINDSET(u),FINDSET(v')); $E' = E' \setminus e$;
        end if
        if FINDSET(v) = FINDSET(v') **then**
            UNION(FINDSET(u),FINDSET(u')); $E' = E' \setminus e$;
        end if
      end if
    end for
  end while
  Return True

6. OPSPANNENDE BOMEN (2): Geef een snel algoritme voor het vinden van een opspannende boom met maximaal gewicht. Leg uit.

7. FIBONACCI HEAPS (2): Wat is de geamortizeerde kost van de DELETE-MIN en DECREASE-KEY operatie wanneer we de potentiaal functie $2t(H) + 3m(H)$ gebruiken? Werk uit.

Veel succes.