

OPEN BOEK EXAMEN

DATASTRUCTUREN EN GRAAFALGORITMEN

16/6/2021

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

1. GRAAF SEARCHING (3): Op hoeveel verschillende manieren kan je de knopen van de graaf $G = (V, E)$ topologisch sorteren indien $V = \{s, u_1, u_2, v_1, v_2, v_3, t\}$ en $E = \{(s, u_1), (u_1, u_2), (u_2, t), (s, v_1), (v_1, v_2), (v_2, v_3), (v_3, t)\}$? Leg je antwoord duidelijk uit.
2. GRAAF SEARCHING (2): Geef twee grafen $G_1 = (V_1, E_1)$ en $G_2 = (V_2, E_2)$ zodat (i) G_1 precies k_1 SCCs bevat, (ii) G_2 precies k_2 SCCs heeft, (iii) er bestaat $u_1, u_2 \in V_1$ en $v_1, v_2 \in V_2$ zodat $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{(u_1, v_1), (v_2, u_2)\})$ slechts één SCC bevat.
3. FLOW NETWERKEN (3): Wat is het maximum aantal minimale cuts in een flow network bestaande uit n knopen? Geef een voorbeeld van een flow netwerk waarvoor het aantal minimale cuts gelijk is aan je antwoord.
4. DISJOINT SETS (3): Stel dat we enkel PATH COMPRESSION gebruiken en geen UNION-BY-RANK. Geef dan een reeks van $n = 10$ MAKESET, 9 UNION en 10 FINDSET operaties zodat het aantal parent pointers dat je moet aanpassen tijdens de FINDSET operaties zo groot mogelijk is. De FINDSET operaties die deel uitmaken van de UNION tellen we niet mee. Teken je datastructuur voor elke FINDSET en nummer je knopen zodat het duidelijk is welke pointers aangepast zijn.
5. OPSPANNENDE BOMEN (2): Geef een graaf G met 4 knopen en 6 edges met gewichten 1, 2, 3, 4, 5 en 6 zodat het gewicht van de opspannende boom met minimaal gewicht zo groot mogelijk is.
6. FIBONACCI HEAPS (4): Stel dat we geen markeringen gebruiken en als potentiaal functie $\Phi(H) = t(H)$. Wat is dan de geamortiseerde kost van een DELETEMIN en DECREASEKEY operatie? Waarom is de complexiteit minder goed dan met markeringen? Leg uit.
7. KORTSTE PADEN (3): Geef een snel algoritme dat voor elke knoop $v \in V$ een zo kort mogelijke cycle C_v zoekt zodat v gelegen is op deze cycle (waarbij de lengte van de cycle gelijk is aan het aantal edges die deel uitmaken van de cycle). Wat is de complexiteit van je algoritme?

Veel succes.