2e deeltentamen Algoritmiek 2019

Je hebt drie uur voor dit deeltentamen.

Schrijf op elk van de ingeleverde bladen je naam en studentnummer, en op het eerste blad het aantal ingeleverde bladen.

Geef duidelijke antwoorden in helder en correct Nederlands of Engels. Wanneer niet expliciet anders gevraagd, mag je resultaten uit het college gebruiken.

Als een algoritme wordt gevraagd is het een goed idee om dit algoritme zowel te geven in pseudocode, als een toelichting in woorden te geven.

Schrijf netjes, en lever overzichtelijk werk in.

Deel je tijd goed in: sommige opgaven kunnen moeilijker zijn dan andere.

1. Minimum opspannende bomen ($7 = 2 + 3 + 2$ punten)

Je bent gegeven een samenhangende graaf $G = (V, E)$ met gewichtsfunctie $w : E \to \mathbb{N}$ op de kanten.

(a) **Leg kort uit hoe het algoritme van Prim-Jarnik voor het vinden van een minimum opspannende boom werkt.**

(b) Stel dat alle gewichten uniek (dus verschillend) zijn. **Toon aan dat de graaf een unieke minimum opspannende boom T heeft.**

(c) Neem weer aan dat alle gewichten uniek zijn. Een *tweed-beste minimum opspannende boom* is een opspannende boom van G die na T het kleinste gewicht heeft. Dus als $\mathcal{T}$ de verzameling opspannende bomen van G is en T de minimum opspannende boom, dan is een tweed-beste minimum opspannende boom dus een boom in $\mathcal{T} \setminus \{T\}$ die minimum totaal gewicht heeft. **Geef een voorbeeld van een graaf met unieke (verschillende) gewichten die niet een unieke tweed-beste minimum opspannende boom heeft.**

2. Queue (10 punten)

Je hebt een nieuwe queue-datastructuur ontworpen: de multi-queue. Deze datastructuur heeft drie operaties:

- `enqueue`(x): dit zet x in de queue.
- `dequeue`(): dit haalt het oudste element uit de queue en print het.
- `dequeue_all`(): dit haalt alle elementen uit de queue en print deze.

Je mag aannemen dat de queue onderliggend door middel van een linked list geïmplementeerd is en dat deze initieel leeg is. **Gebruik de potentiaalmethode om aan te tonen dat de geamortiseerde kosten van ieder van deze operaties in een serie operaties $O(1)$ is.**

3. Failliet ($10 = 8 + 2$ punten)

Een ziekenhuis in het midden van het land gaat failliet. Jij moet de n patiënten van het ziekenhuis verdelen over de k ziekenhuizen die nog wel operationeel zijn. Om de ziekenhuizen niet te overbelasten, kan ieder ziekenhuis hooguit $\lceil n/k \rceil$ patiënten opnemen. Sommige ziekenhuizen kunnen door jarenlange bezuinigingen helaas niet alle behandelingen uitvoeren. Een patiënt i kan daarom alleen naar de verzameling ziekenhuizen Z_i.

(a) **Beschrijf en geef pseudo-code voor een efficiënt algoritme dat uitrekent of het mogelijk is om iedere patiënt naar een ziekenhuis te sturen waar die geholpen kan worden, zodat de ziekenhuizen niet overbelast zijn.** Je algoritme hoeft als antwoord alleen MOGELIJK of ONMOGELIJK terug te geven.

(b) **Analyseer de looptijd van je algoritme.**

4. Koppelingen en Vertex Cover ($13 = 2 + 4 + 2 + 1 + 4$ punten)

(a) **Leg uit wat het verschil is tussen een maximale en een maximum koppeling.**

(b) **Geef een algoritme dat in lineaire tijd een maximale koppeling vindt in een algemene graaf. Analyseer de correctheid en looptijd van je algoritme.**

Herinner dat een *vertex cover* in een graaf $G = (V, E)$ een verzameling $C \subseteq V$ van knopen is zodanig dat voor iedere kant $(u, v) \in E$ geldt dat $u \in V$ of $v \in V$ (u en v mogen ook allebei in V zitten.

In bovenstaande figuur vormen de zwarte knopen een vertex cover.

(c) **Laat zien dat de kleinste (minimum) vertex cover van een graaf G tenminste zoveel knopen bevat als het aantal kanten in een maximale koppeling.**

(d) Bekijk een maximale koppeling M in een graaf G. **Laat zien dat de uiteinden van de kanten in M een vertex cover van G vormen.**

(e) **Laat zien hoe je het algoritme uit deelvraag (b) kunt gebruiken om een efficiënt 2-benaderingsalgoritme voor het vinden van een kleinste (minimum) vertex cover in algemene grafen te geven. Analyseer de looptijd en bewijs de benaderingsfactor van je algoritme.**

5. Kortste paden ($10 = 2 + 3 + 3 + 1$ punten)

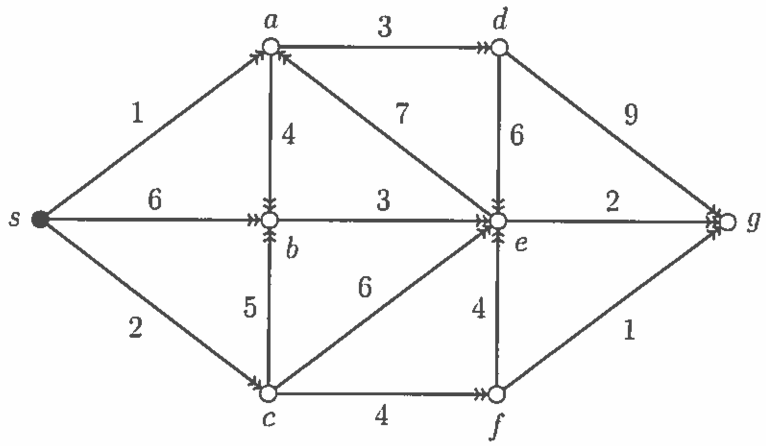

Figure 1: Graaf bij Opgave 5(a).

(a) Gegeven de graaf in Figuur 1. **In welke volgorde neemt Dijkstra's algoritme, gestart vanaf s, de knopen in de wolk op?** Met andere woorden, geef de volgorde van de knopen aan waarin Dijkstra's algoritme de afstand tot een knoop zeker weet.

(b) Gegeven is een gerichte graaf $G = (V, A)$ met n knopen en m pijlen van niet-negatieve integer lengte, en een knoop s. Ook gegeven is een array B van niet-negatieve integers, waarvan wordt beweerd dat $B[v]$ de correcte afstand van s naar knoop v is, voor iedere $v \in V$. **Geef een algoritme met looptijd $O(n + m)$ dat besluit of alle waarden $B[v]$ correct zijn.** Je mag aannemen dat alle knopen vanuit s bereikbaar zijn.

(c) **Leg (kort) uit waarom je algoritme correct is.**

(d) **Leg (kort) uit waarom je algoritme looptijd $O(n + m)$ heeft.**