Hertentamen Algoritmiek 3 januari 2003

U heeft 3 uur de tijd voor dit tentamen. Schrijf Uw naam op alle blaadjes die U inlevert. Bij elk onderdeel staat het aantal punten dat U voor dit onderdeel kunt halen aangegeven. U kunt in het totaal 50 punten halen; Uw onafgeronde eindcijfer is het aantal punten gedeeld door 5. Veel succes!

Opgave 1:

(5 punten) Gegeven is een graaf $G = (V, E)$ met voor elke kant $e \in E$, een lengte $\ell(e)$. Lengtes zijn positieve integers. Gegeven is ook een tweetal knopen $v \in V$, en $w \in W$. We willen weten wat de lengte van het kortste pad van v naar w is.

Welk algoritme kan dit probleem in $O(|V|^2)$ tijd oplossen? Leg compact uit hoe dit algoritme werkt.

Opgave 2:

Een elektronische schakeling bestaat uit een aantal componenten $C_1, C_2, \ldots, C_n$. De elektronische schakeling is ontworpen, zodat alle n componenten in **een enkele rij** geplaatst worden (maar in een willekeurige volgorde). De componenten kunnen door 0, 1 of meer draden verbonden zijn. Laat $W(C_i, C_j)$ het aantal draden zijn waarmee component C_i verbonden is met component C_j. Gegeven zijn het aantal componenten, en voor elk paar componenten C_i, C_j de waarde van $W(C_i, C_j)$ (een niet-negatief geheel getal.) Merk op: $W(C_i, C_j) = W(C_j, C_i)$.

We willen een volgorde van de componenten vinden die de kosten van de verbindingen minimaliseert (met andere woorden: we willen zo weinig mogelijk draad gebruiken). De kosten om twee componenten te verbinden zijn

Kosten$(C_i, C_j) = W(C_i, C_j) * (1 +$ aantal componenten tussen C_i en $C_j)$

De totale kosten zijn gelijk aan de som van de kosten over alle paren componenten, d.w.z., $\sum_{i,j,i \neq j}$ Kosten(C_i, C_j).

Stel bijvoorbeeld dat $n = 4$, en we hebben de volgende hoeveelheden verbindingen:

$W(C_1, C_2) = 8$, $W(C_1, C_3) = 6$, $W(C_1, C_4) = 0$, $W(C_2, C_3) = 3$, $W(C_2, C_4) = 4$, $W(C_3, C_4) = 10$. Natuurlijk is $W(C_1, C_1) = W(C_2, C_2) = W(C_3, C_3) = W(C_4, C_4) = 0$.

Als we de componenten in de volgorde C_1, C_3, C_4, C_2 plaatsen, dan zijn de totale kosten:

$$3 * W(C_1, C_2) + 1 * W(C_1, C_3) + 2 * W(C_2, C_3) + 1 * W(C_2, C_4) + 2 * W(C_3, C_4) = 50$$

Om de kosten te minimaliseren zullen we in het algemeen de componenten die door veel draden verbonden worden, dicht bij elkaar moeten plaatsen.

(a) (3 punten) Ontwerp een greedy algoritme om dit probleem op te lossen. Beschrijf het algoritme in pseudocode.

(b) (2 punten) Toon aan dat je greedy algoritme altijd optimale volgordes oplevert, of geef een tegenvoorbeeld.

1

(c) (5 punten) Ontwerp een backtracking algoritme dat altijd een optimale plaatsing van componenten vindt. Zorg ervoor dat een partiële plaatsing, die altijd slechter is dan de beste plaatsing die al gevonden is, door het algoritme niet verder wordt geanalyseerd. Beschrijf het algoritme in pseudocode.

Opgave 3:
Deze vraag gaat over de netwerkstroomgraaf en de stroom uit figuur 1.
(a) (1 punt) Is de stroom optimaal?

(b) (4 punten) Geef in het geval dat de stroom optimaal is een snede aan met dezelfde capaciteit als de stroom. Geef als dit niet het geval is, een toevoegend pad aan.

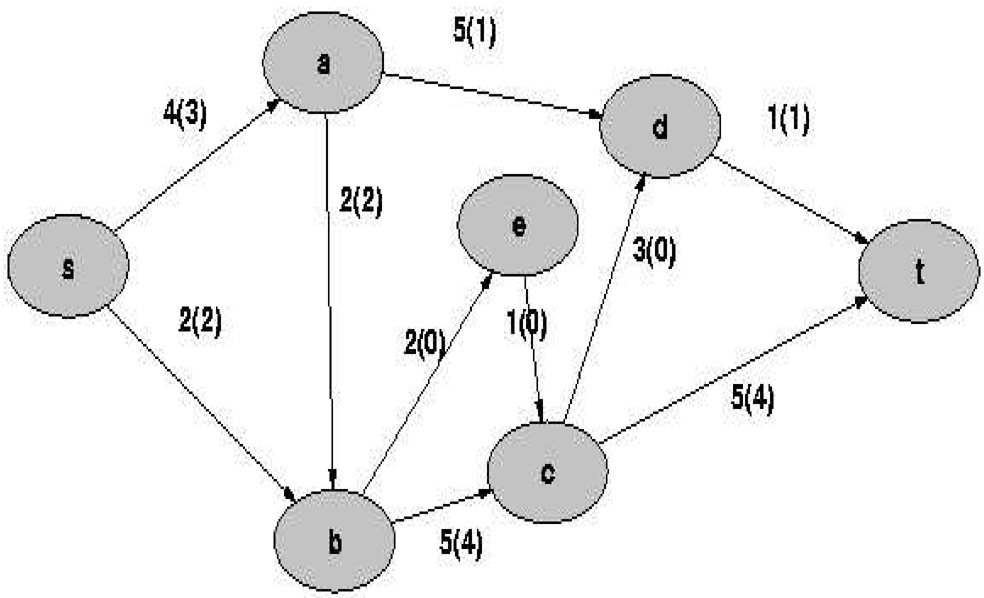

Figure 1: Netwerkstroomgraaf. Een label zoals 5(4) bij een pijl betekent dat de capaciteit van de pijl 5 is en de stroom 4 is.

Opgave 4:
(5 punten) Stel een ongerichte graaf $G = (V, E)$ is gegeven door middel van de adjacency matrix datastructuur. Stel G heeft $n^2/8$ kanten. Geef een Las Vegas algoritme dat in $O(1)$ verwachte tijd een paar knopen dat door een kant verbonden is vindt. Beargumenteer de tijdgrens.

Opgave 5: Dynamisch programmeren
(a) 5 punten. Anne en Fleur spelen het volgende spelletje (Nim). Er zijn drie stapels lucifers, met n_1, n_2, en n_3 lucifers. Om beurten mag een speler één of meer lucifers van een stapel pakken. De speler die de allerlaatste lucifer pakt wint het spel. Anne begint.

Doel van deze opgave is het maken van een dynamisch programmeeralgoritme om te bepalen wie een winnende strategie voor dit spel heeft. We schrijven: $W(q, r, s) =$ true, dan en slechts dan als de beginnende speler een winnende strategie heeft wanneer de stapels respectievelijk q, r, en s lucifers bevatten. De volgende recurrente betrekking kan voor W worden opgesteld:

- $W[0,0,0] = \text{false}$
- $W[0,0,s] = \text{true}$, als $s > 0$.
- $W[0,r,0] = \text{true}$, als $r > 0$.
- $W[q,0,0] = \text{true}$, als $q > 0$.
- Als $q > 0$, $r > 0$, en $s > 0$, dan:

$$W[q,r,s] = \left(\bigvee_{0 \le t < q} \neg W(t,r,s) \right) \vee \left(\bigvee_{0 \le t < q} \neg W(q,t,s) \right) \vee \left(\neg \bigvee_{0 \le t < q} W(q,r,t) \right)$$

Geef een dynamisch programmeer algoritme dat, gebruik makend van bovenstaande recurrente betrekkingen, bepaalt of de beginnende speler een winnende strategie heeft, wanneer de stapels n_1, n_2, n_3 lucifers hebben, voor gegeven n_1, n_2, n_3. De tijd van Uw algoritme moet begrensd zijn in een polynoom in $\max\{n_1, n_2, n_3\}$.

(b) 1 punt. Geef een afschatting van de tijd die Uw algoritme gebruikt, als functie van $\max\{n_1, n_2, n_3\}$, of als functie van n_1, n_2, n_3.

(c) 5 punten. Gegeven zij een verzameling positieve gehele getallen $\{a_1, \ldots, a_n\}$, en een positief geheel getal B. We willen bepalen hoeveel deelverzamelingen van $\{a_1, \ldots, a_n\}$ er zijn met som precies B.

Voor $0 \le i \le n$, $0 \le C \le B$, definiëren we $X(i, C)$ als het aantal deelverzamelingen van $\{a_1, \ldots, a_i\}$ met som precies C.

Bijvoorbeeld: als $n = 5$, $a_1 = 10$, $a_2 = 20$, $a_3 = 30$, $a_4 = 40$, $a_5 = 15$, dan is $X(5, 50) = 2$ (want 50 kan op twee manieren gekregen worden als een som van een deelverzameling van $\{10, 20, 30, 40, 15\}$. $X(2, 25) = 0$; $X(5, 25) = 1$.

Geef een recursieve definitie (recurrente betrekking) voor X.

Opgave 6 (5 punten) Laat $T[1 \ldots n]$ een array zijn met n integers. Geef een algoritme dat bepaalt of er een integer is die tenminste $n/3$ keer in het array voorkomt. Uw algoritme moet $O(n)$ tijd gebruiken.

Als dit U niet lukt, geef dan een algoritme dat $O(n \log n)$ tijd gebruikt. (Dit geeft maximaal 2 punten.)

Beargumenteer Uw tijdgrens.

Opgave 7

(a) (1 punt.) Noem twee NP-volledige problemen.

(b) (1 punt.) Noem een probleem dat tot de klasse P behoort.

(c) (3 punten.) Zeg van elk van de onderstaande beweringen of ze juist zijn, en beargumenteer Uw antwoord kort.

1. Wanneer een probleem NP-volledig is, dan is er waarschijnlijk geen algoritme dat het probleem altijd in polynomiale tijd exact oplost.

2. Wanneer een probleem tot de klasse NP behoort, dan is het niet mogelijk dat het probleem tot de klasse P behoort.

3. Als een probleem een polynomiale tijd approximatie schema heeft, dan is het niet NP-volledig, tenzij $P = NP$.

Opgave 8 (4 punten) Het Bottleneck path probleem is het volgende. Gegeven is een graaf gerichte graaf $G = (V, E)$, met voor elke pijl $(x, y) \in E$ een gewicht $w(x, y) \in \mathbf{Z}$. Gegeven zijn ook twee knopen $v \in V$, $w \in V$. Gezocht wordt een pad van v naar w waarvan het *maximum gewicht* van een pijl op het pad zo klein mogelijk is, Als er helemaal geen pad van v naar w is in G, dan wordt natuurlijk geen pad opgeleverd, maar wordt een passende foutmelding gegeven.

Leg uit hoe dit probleem *efficient* kan worden opgelost.