Departement Informatica en Informatiekunde, Faculteit Bètawetenschappen, UU.
In elektronische vorm beschikbaar gemaakt door de $\mathcal{TBC}$ van A−Eskwadraat.
Het college INFOAL werd in 2004/2005 gegeven door Hans Bodlaender.

Algoritmiek (INFOAL)
24 maart 2005

Opgave 1. Koppelingen *(1.5 punten)*

In een bedrijf zijn n werknemers $p_1, \ldots, p_n$ en n taken $t_1, \cdots, t_n$. Elke werknemer heeft voor sommige taken wel de vaardigheid om die uit te voeren, maar voor andere taken niet. We willen iedere taak aan één werknemer toewijzen, zodat elke werknemer precies één taak toegewezen krijgt. Het probleem is om te bepalen of zo'n toewijzing bestaat, en zo ja, die op te leveren.
Het probleem kan gemodelleerd worden als het vinden van een koppeling in een bipartite graaf $G = (V_1 \cup V_2, E)$. Leg uit hoe dit probleem kan worden opgelost met behulp van een algoritme dat een maximum stroming in een graaf berekent. Leg duidelijk uit hoe de graven eruit zien waarin het maximum stromingsalgoritme werkt.

Opgave 2. Benaderingsalgoritmen en NP-volledigheid *(3 punten)*

Een verzameling knopen W is onafhankelijk (*independent*) in een graaf $G = (V, E)$ als voor alle $v, w \in W : \{v, w\} \notin E$. Een *maximum onafhankelijke deelverzameling* in een graaf $G = (V, E)$ is een onafhankelijke deelverzameling W met $|W|$ zo groot mogelijk.
We willen nu het volgende probleem bekijken:

Gegeven: Graaf $G = (V, E)$, met voor elke $v \in V$: de graad van v is hooguit 4.

Gevraagd: Geef een maximum onafhankelijke deelverzameling $W \subseteq V$ in G.

a) Er is bewezen dat de beslissingsvariant van dit probleem NP-volledig is. Bespreek wat dit resultaat betekent, en wat de consequenties ervan zijn als we het bovenstaande probleem willen oplossen.

b) Beschouw de volgende heuristiek:

> $X = \emptyset$;
> $W = V$
> **while** $W \neq \emptyset$ **do**
>   Kies een knoop $x \in W$.
> $X = X \cup \{x\}$.
>   Haal x en alle buren van x uit W.
> Output X.

Leg uit dat deze heuristiek een onafhankelijke deelverzameling oplevert.

c) Stel een maximum onafhankelijke verzameling in G heeft formaat OPT. Toon aan det het algoritme een onafhankelijke verzameling van formaaat tenminste OPT/4 geeft.

Opgave 3. Details van Johnson's kortste paden algoritme *(1.5 punten)*

We bekijken een element van Johnson's kortste paden algoritme. Stel we hebben een gerichte graaf $G = (V, E)$, met elke kant $(v, w) \in E$ een lengte $\ell(v, w) \in \mathbb{R}$. Alle knopen $v \in V$ hebben een hoogte $h(v) \in \mathbb{R}$. Definieer een nieuwe lengtefunctie $\ell_h : E \to \mathbb{R}$ met voor alle $(v, w) \in E$: $\ell_h(v, w) + h(v) - h(w)$.

Stel P is een pad in G van s naar t. Stel $\ell(P)$ is de totale lengte van P met lengtefunctie ℓ, en $\ell_h(P)$ is de totale lengte van P als we lengtefunctie ℓ_h gebruiken. Bewijs:

$$\ell_h(P) = \ell(P) + h(s) - h(t).$$

Opgave 4. DFS en toepassingen *(4 punten)*

Resultaten en algoritmen behandeld in het college kunt U als gegeven beschouwen. U hoeft bijvoorbeeld neit de pseudocode van DFS te geven, wanneer U DFS nodig heeft. Wanneer U een onderdeel niet hebt kunnen bewijzen, mag U dat onderdeel toch in een later gedeelte gebruiken.

Eeen *brug* in een ongerichte graaf is een kant $\{v, w\}$ zodat de samenhangende component waar $\{v, w\}$ in zit in twee of meer stukken splitst wanneer de kant $\{v, w\}$ verwijderd wordt. Met andere woorden: het enige pad in de graaf tussen v en w gaat via de kant $\{v, w\}$ verwijderd wordt. Met andere woorden: het enige pad in de graaf tussen v en w gaat via de kant $\{v, w\}$.

We gaan een algoritme bouwen dat de bruggen in een gegeven ongerichte graaf $G = (V, E)$ vindt. G is gegeven door een adjecency list datastructuur.

a) Stel we hebben een DFS-boom in G. Kan een back-edge een brug zijn? Zo ja, geef een voorbeeld. Zo nee, waarom niet?

b) Defineer het BEO (*back-edges overschot*) van een knoop v als het aantal back-edges $\{x, y\}$ met de eigenschap dat één eindpunt van $\{x, y\}$ een voorouder van v is, en één eindpunt een afstammeling van v of v zelf is.

   Laat zien dat, gegeven een DFS-boom van een graaf $G = (E, V)$, we het BEO van elke knoop in $O(|V| + |E|)$ tijd kunnen berekenen.

c) Stel $\{u, v\}$ is een kant in de DFS boom, met u de ouder van v. Toon aan dat $\{u, v\}$ een brug is , dan en slechts dan als het BEO van v 0 is.

d) Beschrijf een algoritme dat alle bruggen van een gegeven ongerichte graaf $G = (V, E)$ vindt. Uw algoritme moet $O(|V| + |E|)$ tijd gebruiken.