Discrete modellen in de toegepaste wiskunde (WISB136). Uitwerkingen van het tentamen.

Docent: Rob H. Bisseling, Universiteit Utrecht

15 april 2013. Tijd 9.00 - 12.00 uur.

1. (a) Een bipartiete graaf $G = (V, E)$ heeft een verzameling knopen $V = U \cup W$ met $U, W \neq \emptyset$ and $U \cap W = \emptyset$, en alle zijden zijn te schrijven als (u, w) met $u \in U$ en $w \in W$.

   (b) Neem aan dat de kubus een eenheidskubus is en nummer de knopen van 0 tot 7, met $0 \equiv (0, 0, 0)$, $1 \equiv (1, 0, 0)$, $2 \equiv (0, 1, 0)$, $3 \equiv (1, 1, 0)$, $4 \equiv (0, 0, 1)$, $5 \equiv (1, 0, 1)$, $6 \equiv (0, 1, 1)$, $7 \equiv (1, 1, 1)$. Plaats knoop 0 in U, dan moeten 1, 2, 4 in W geplaatst worden. Dan mogen 3, 5, 6 niet in W geplaatst worden, dus moeten ze wel in U. Knoop 7 moet dan in W terechtkomen. De knopenverzamelingen moeten dus $U = \{0, 3, 5, 6\}$ en $W = \{1, 2, 4, 7\}$ zijn. Het is makkelijk te controleren dat alle zijden een knoop van U met een knoop van W verbinden.

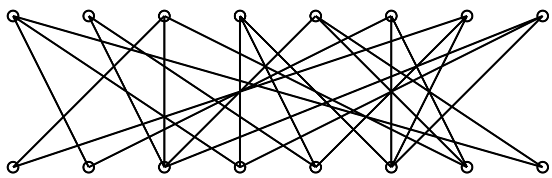
   (c) Een matching M in een graaf G is een deelverzameling van E zodanig dat elke knoop $v \in V$ ten hoogste eindpunt van één zijde $(v, x) \in M$ is.

   (d) Een maximummatching M heeft het maximum aantal zijden.
   $M = \{(0, 4), (1, 5), (2, 6), (3, 7)\}$ heeft 4 zijden. Meer zijden in een matching van 8 knopen kan niet.

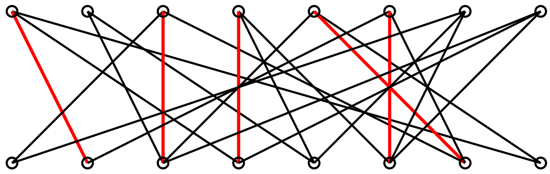
   (e) De graaf Q_3 is planair. Laat knopen 0–3 in het bodemvlak, en bed ze als volgt in het platte vlak: $0 \equiv (0, 0)$, $1 \equiv (1, 0)$, $2 \equiv (0, 1)$, $3 \equiv (1, 1)$. Verplaats 4 naar $(1/4, 1/4)$, 5 naar $(3/4, 1/4)$, 6 naar $(1/4, 3/4)$, 5 naar $(3/4, 3/4)$. Verbind de knopen met rechte lijnstukken. Dit geeft een planaire inbedding.

2. (a) Er mogen geen twee pivots in dezelfde rij voorkomen, want anders zou deze rij naar twee verschillende posities gepermuteerd moeten worden. Net zo mogen er ook geen twee pivots in dezelfde kolom voorkomen. We maken als volgt een bipartiete graaf van de matrix. De rijen zijn knopen $r_1, \ldots, r_8$; de kolommen knopen $k_1, \ldots, k_8$. Bij elke $a_{ij} \neq 0$ hoort een zijde $(r_i, k_j) \in E$. Bij elke rij i hoort ten hoogste één pivot, en dus ten hoogste één kolom j. Bij elke rijknoop r_i hoort dus ten hoogste één zijde $(r_i, k_j) \in P$. Net zo voor kolommen. P is dus een matching. Het probleem is dus een maximummatching P te vinden voor deze bipartiete graaf.

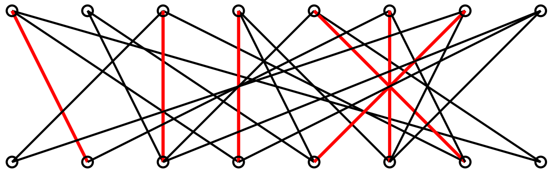
   (b) De graaf kan als volgt getekend worden, met boven de rijknopen en beneden de kolomknopen, genummerd van links naar rechts:



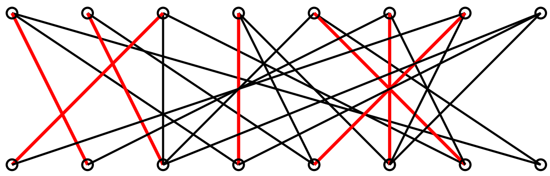
(c) Dit grafenprobleem kan opgelost worden met het Hongaarse algoritme dat APS-bomen maakt, met hopelijk augmenterende paden. (Voor het goed zien van de rode lijnen in de volgende plaatjes is printen in kleur of kijken naar een kleurenscherm essentieel.) We kunnen starten met de volgende eenvoudig te vinden matching



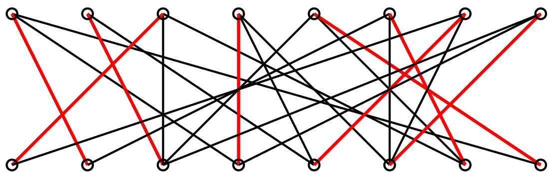
We proberen nog meer paren te matchen door eerst (r_7, k_5) toe te voegen:



Er is een augmentered pad r_2, k_3, r_3, k_1, dat we omklappen zodat (r_2, k_3) en (r_3, k_1) in de matching komen:



Er is een augmentered pad $k_8, r_5, k_7, r_6, k_6, r_8$ dat we omklappen:



Er zijn dus 8 pivots mogelijk. De gevraagde verzameling is:
$P = \{a_{12}, a_{23}, a_{31}, a_{44}, a_{58}, a_{67}, a_{75}, a_{86}\}$.

3. (a) Een Eulerse graaf is een graaf $G = (V, E)$ waarvoor een Eulertoer bestaat, d.w.z. een pad $v_1 v_2 \cdots v_r$ met $v_r = v_1$, $r \geq 1$, en $v_i v_{i+1} \in E$ voor $i = 1, \ldots, r-1$, waarbij elke zijde in de graaf precies éénmaal doorlopen wordt.

(b) Alle knopen van de linkergraaf hebben een even graad en de graaf is samenhangend, dus er is een Eulertoer. Als we de knopen op de eerste rij 1, 2, 3 nummeren, die op de volgende rij 4, 5, 6, enz., en die op de laatste rij 13, 14, dan is een Eulertoer: 1, 4, 7, 11, 10, 13, 14, 11, 12, 8, 9, 6, 8, 7, 5, 3, 2, 4, 5, 6, 1.

De rechtergraaf heeft knopen met oneven graad (bijv de vierde knoop op de tweede rij van boven heeft graad 5), dus er bestaat geen Eulertoer.

4. (a) Als we de knopen op de eerste rij 1, 2, 3 nummeren, die op de volgende rij 4, 5, 6, en die op de laatste rij 7, 8, dan zien we dat we met drie kleuren kunnen volstaan: Rood 2, 6, 7; geel 3, 4, 8; Blauw 1, 5. Door te kijken naar driehoek 1-2-4 zie je dat er minstens drie kleuren nodig zijn, want deze drie knopen zijn allen met elkaar verbonden.

(b) (i) Stel een graaf is met 2 kleuren te kleuren, en we hebben een cykel d.w.z. een pad $v_1 v_2 \cdots v_r$ met $v_r = v_1$, $r \geq 1$, en $v_i v_{i+1} \in E$ voor $i = 1, \ldots, r-1$. Stel de lengte van de cykel, $r-1$, is oneven. Neem een 2-kleuring van de graaf, zeg met de kleuren rood en blauw. Stel de kleur van v_1 is blauw, dan moet v_2 rood zijn, vanwege zijde $v_1 v_2$, en daarna moet v_3 weer blauw zijn, enz. Op die manier zien we dat alle even knopen rood zijn en de oneven knopen blauw. Echter, r is even en v_r is dus blauw, maar vanwege $v_1 = v_r$ is deze knoop ook rood. Tegenspraak, en de lengte van de cykel moet wel even zijn.

(ii) Stel elke cykel van een graaf G heeft even lengte. Kleur de graaf als volgt met de kleuren rood en blauw. Neem een willekeurige startknoop en kleur die blauw. Vervolgens kleur alle buren van die rode knoop blauw en de buren van die blauwe knopen weer rood. Dit zetten we voort tot er geen ongekleurde buren meer zijn. Op dat moment hebben we een samenhangende component van G geheel gekleurd. Als we zo dezelfde knoop twee keer tegenkomen, en dus twee keer kleuren, hebben we twee paden naar die knoop. Als we die paden met elkaar verbinden krijgen we een cykel. Die heeft even lengte, en de twee paden hebben dus beide even lengte of beide oneven lengte. In het even geval is het eindpunt van het pad een blauwe knoop, en in het oneven geval een rode knoop. De betreffende knoop wordt dus twee keer in dezelfde kleur gekleurd. Als de graaf niet samenhangend is, wordt elke component apart gekleurd volgens bovenstaande methode. Zo krijgen we dus een 2-kleuring.

5. (a) Het algoritme van Dijkstra voor het vinden van een kortste pad vanuit een startpunt S $\in V$ in een gerichte graaf $G = (V, E)$ is als volgt. Neem aan dat er een directe verbinding bestaat met lengte $l(v, w) \geq 0$ voor elke zijde van v naar w. Dit algoritme berekent de kortste afstand $\mathit{dist}(v)$ van S naar elke knoop $v \in V$ en dus ook naar T. Voor elke knoop v wordt ook de huidige voorganger $\mathit{pred}(v)$ opgeslagen. D is de verzameling knopen waarvoor we de kortste afstand al gevonden hebben.

for all $v \in V$ **do**
    $\mathit{dist}(v) \leftarrow \infty$; $\mathit{pred}(v) \leftarrow v$;
$\mathit{dist}(S) \leftarrow 0$;
$D \leftarrow \emptyset$;
while $D \neq V$ **do**
    zoek $v \in V \backslash D$ met minimale $\mathit{dist}(v)$;
    $D \leftarrow D \cup \{v\}$;
    for all $w \in V \backslash D \ : \ (v, w) \in E$ **do**
        if $\mathit{dist}(v) + l(v, w) < \mathit{dist}(w)$ **then**
            $\mathit{dist}(w) \leftarrow \mathit{dist}(v) + l(v, w)$;
            $\mathit{pred}(w) \leftarrow v$;

(b) De knopen v worden in deze volgorde gekozen: S, A, C, B, T. De respectieve definitieve afstanden vanaf S zijn: 0, 1, 3, 5, 7. Het kortste pad is S, A, C, B, T en heeft lengte 7.