Freie Universität Berlin
Institut für Mathematik
Algorithmic Combinatorics
Lecturers: Tibor Szabó
Anurag Bishnoi

Winter Semester 2018-19

| Exercise | 1. | 2. | 3. | 4. | 5. | 6. |
|---|---|---|---|---|---|---|
| Points | | | | | | |

# PRACTICE EXAM - solution

**Exercise 1** [10 points]

(1) *Dijkstra's Algorithm.*
**Input**: Graph $G = (V, E)$, weight function $w : E \to \mathbb{R}_{\geq 0}$, an initial vertex $u \in V$.

**Output**: For all $v \in V$, $D(v)$ = shortest distance from $u$ to $v$.

**Initialization**: Extend $w$ so that $w(xy) = \infty$ for all $xy \in \binom{V}{2} \setminus E$. Maintain a list $D$ of shortest distances from $u$ indexed by the vertices and put $D(u) = 0$, $D(v) = \infty$ for all $v \in V \setminus \{u\}$. Maintain a list $W$ of visited vertices and start with $W = \emptyset$.

**Iteration**:

- Let $v_0 = \text{argmin}\{D(v) : v \in V \setminus W\}$.
- Update $W = W \cup \{v_0\}$.
- For all $v \in V \setminus W$, IF $D(v_0) + w(v_0 v) < D(v)$, THEN update $D(v) = D(v_0) + w(v_0, v)$.

**Termination**: Stop if $W = V$ and output $D$.

(2) At the initialisation stage we have $W = \emptyset$ and $D = (0, \infty, \ldots, \infty)$ where the order of the vertices is $a, b, c, d, e, f$. Then we have the following iterations.

1. $v_0 = a$, $W = \{a\}$, $D = (0, 2, 4, \infty, \infty, \infty)$.
2. $v_0 = b$, $W = \{a, b\}$, $D = (0, 2, 3, 6, 4, \infty)$.
3. $v_0 = c$, $W = \{a, b, c\}$, $D = (0, 2, 3, 6, 4, \infty)$.
4. $v_0 = e$, $W = \{a, b, c, e\}$, $D = (0, 2, 3, 6, 4, 6)$.
5. $v_0 = d$, $W = \{a, b, c, e, d\}$, $D = (0, 2, 3, 6, 4, 6)$.
6. $v_0 = f$, $W = \{a, b, c, e, d, f\}$, $D = (0, 2, 3, 6, 4, 6)$.

Return $D = (0, 2, 3, 6, 4, 6)$.

**Exercise 2** [10 points]

(1) A decision problem is in NP if for all the instances $I$ of that problem where the answer is YES, there exists a polynomial time algorithm that proves that the answer for $I$ is yes (possible with the help of a non-deterministic oracle). A decision problem is NP-complete if it is in NP and every NP problem has a polynomial time reduction to the problem.

(2) The $k$-SAT problem is in NP, because if the $k$-CNF formula is satisfiable, the proof is just the value assignment, which can be checked in time proportional to the number of clauses. We will give a polynomial time reduction of 3-SAT to $k$-SAT to show that $k$-SAT is NP-complete, since every NP problem can then be reduced to a $k$-SAT by first reducing it to 3-SAT and then using this reduction.

Let $f(x_1, \ldots, x_n)$ be a 3-CNF with clauses $C_1, \ldots, C_m$. Introduce $k-3$ new variables, $y_1, \ldots, y_{k-3}$ and let $D_1, \ldots, D_{2^{k-3}}$ be the all the $2^{k-3}$ clauses of length $k-3$ we can make out of these variables. Consider the $k$-CNF,

$$g(x_1, \ldots, x_n, y_1, \ldots, y_{k-3}) = (\wedge_{i=1}^{2^{k-3}}(C_1 \vee D_i)) \wedge \cdots \wedge (\wedge_{i=1}^{2^{k-3}}(C_m \vee D_i)).$$

Then $g$ has $2^{k-3}m$ clauses. We claim that $f$ is satisfiable if and only if $g$ is satisfiable.

Say $f$ is satisfiable. Then each $C_i$ has a literal that has value $T$, and hence each $C_i \vee D_j$ has a literal that has value $T$. This implies that $g$ is also satisfiable. Now say $f$ is not satisfiable, and take any assignment of the variables $x_1, \ldots, x_n, y_1, \ldots, y_{k-3}$. No matter what the assignment of $y_1, \ldots, y_{k-3}$ there exists an $j$ such that $D_i$ is false on these values[1]. Since $f$ is unsatisfiable, there exists an $i$ such that $C_i$ is false. Therefore, $C_i \vee D_j$ is false in $g$, and hence $g$ is false.

This reduction is polynomial time because the number of variables in $g$ is $n+k-3$ and the number of clauses and $2^{k-3}m$, and $k$ is fixed.

**Exercise 3** [10 points]

(1) From Tutte's theorem, it suffices to show that for every $S \subseteq V(G)$, $|S| \geq o(G \backslash S)$ where $o(G \setminus S)$ is the number of connected components in the graph $G[V \setminus S]$. Let $S \subseteq V(G)$, and let $C$ be an odd component of $G[V \setminus S]$. The number of edges between $S$ and $C$ is at least $\kappa'(G) > 1$, since otherwise we will have less than $\kappa'(G)$ edges of $G$ whose removal disconnects the graph. By summing up the degrees of the vertices in $C$, and using the 3-regularity of the graph, we have

$$3|V(C)| = \sum_{v \in G} \deg_G(v) = |[S, V(C)]| + 2E(C),$$

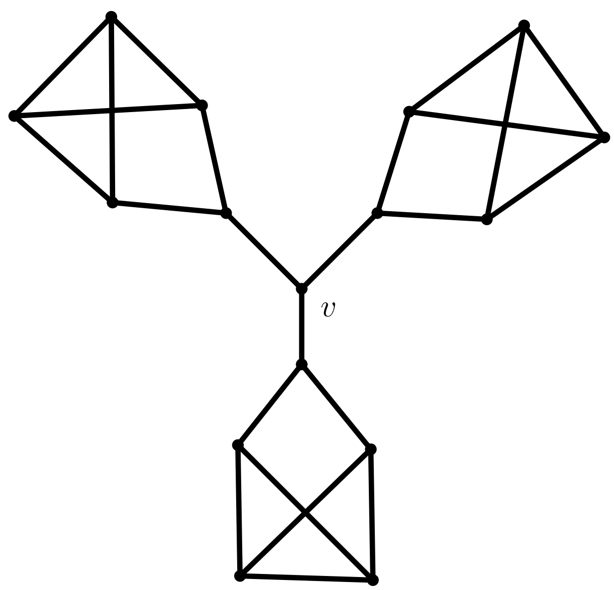
where $[S, V(C)]$ is the set of edges with one end point in $S$ and another end point in $V(C)$. Here we have used the Handshake Lemma in the second equality. This implies that $|[S, V(C)]| = 3|V(C)| - 2E(C)$ is an odd number, and hence at least

[1] we did this in a homework

equal to 3 (since it can't be equal to 1 as we showed above). By summing up the degrees of vertices in $S$, as before, we have

$$3|S| = \sum_{v \in S} \deg_G(v) \geq \sum_{C \text{ is an odd componenet}} |[S, V(C)]| \geq 3o(G \setminus S).$$

Therefore, $|S| \geq o(G \setminus S)$.




(2) The removal of $v$ leaves three odd components, and hence this 3-regular graph has no perfect matchings by Tutte's theorem.

**Exercise 4** [10 points]

(1) Given $f : V(G) \to 2^{\mathbb{N}}$, $G$ is called $f$-colorable if there exists $c : V(G) \to \mathbb{N}$, such that $c(v) \in f(v)$for all $v \in V(G)$, and if $uv \in E(G)$, then $c(u) \neq c(v)$.

$$\chi_l(G) = \min\{k \in \mathbb{N} : G \text{ is } f\text{-colorable}, \forall f : V(G) \to 2^{\mathbb{N}} \text{ with } f(v) \geq k, \forall v \in V(G)\}.$$

(2) We will prove the following Lemma from which we will derive the main result.

**Lemma 1.** *Let $G$ be a plane graph in which the outer face is a cycle $C = v_1, \ldots, v_k$ and all the other faces are triangles. Assume that $v_1, v_2$ are coloured 1 and 2, respectively, for every vertex $v$ on $C \setminus \{v_1, v_2\}$ we have a list $L(v)$ of at least 3 colours, and for every vertex $v$ in $G \setminus C$ we have a list $L(v)$ of at least five colours. Then $G$ can be properly list coloured with these lists.*

*Proof.* We apply induction on the number of vertices. For $n = 3$, we have a triangle for which this result holds because $v_3$ has a colour other than 1 and 2 available for it. Now assume that the results holds true for all graphs with up to $n - 1$ vertices and let $C = v_1, \ldots, v_k$ be the outerface of an $n$ vertex graph $G$, as in the Lemma.

**Case 1**: $C$ has a chord $v_iv_j$, where $v_i$ and $v_j$ are non-adjacent and $2 \leq i < j$ (this can be assumed without loss of generality). Let $G_1$ be the graph containing the cycle $C_1 = v_1, \ldots, v_i, v_j, v_{j+1}, \ldots, v_k, v_1$ and its interior vertices. It satisfies the condition of the Lemma above, and has less vertices than $G$. Therefore by induction hypothesis we can properly colour it. Once we have obtained this colouring, $v_i$ and $v_j$ receive a colour. We can then apply the induction hypothesis to the graph $G_2$ consisting of the cycle $v_i, v_{i+1}, \ldots, v_j, v_i$, and its interior, with $v_i$ and $v_j$ playing the role of $v_1$ and $v_2$. This gives us a proper colouring of the whole graph.

**Case 2**: $C$ has no chords. Let $v_1, u_1, u_2, \ldots, u_m, v_{k-1}$ be the neighbours of $v_k$ in the clockwise order (assuming that $v_1, \ldots, v_k$ are also in a clockwise order). Since all the bounded faces are triangles, there must be a path $P = v_1, u_1, \ldots, u_m, v_{k-1}$ in $G$. Since $G$ has no chords, $P \cup (C \setminus \{v_k\})$ is a cycle $C'$. Let $c_1, c_2$ be two distinct elements of the set $L(v_k) \setminus \{1\}$. For all $1 \leq i \leq m$, define $L'(u_i) = L(u_i) \setminus \{c_1, c_2\}$, and if $L'(u_i)$ still has size more than 3 elements, then remove some to make sure that $|L'(u_i)| = 3$ for all $i$. For all $v \in V(G) \setminus \{u_1, \ldots, u_m\}$ define $L'(v) = L(v)$. By the induction hypothesis the graph consisting of $C'$ and its interior vertices now has a proper list colouring $c'$ with $L'$ as the list assignment. We can then colour $v_k$ using an element of $L(v_k) \setminus \{1, c'(v_{k-1})\}$, which has size 1, to get a proper colouring of $G$.

□

Now given a planar graph $G$, and some lists of colours on its vertices, take a planar embedding of $G$, triangulate it (that is, keep adding edges to it until it stays planar). Then we are in the setting of the Lemma above, and thus this larger graph can be properly coloured using these list which implies that $G$ can be properly coloured.

**Exercise 5** [10 points]

(1) $\chi'(G)$ is the minimum $k$ for which $G$ has a proper edge colouring with $k$ colours, where a properly colouring is a map $c : E(G) \to [k]$ such that for all $e \neq f \in E(G)$ with $e \cap f \neq \emptyset$, we have $c(e) \neq c(f)$. Vizing's theorem says that for a simple graph $G$, we have $\chi'(G) \leq \Delta(G) + 1$.

(2) Since $G$ is a $d$-regular graph, we know that $\chi'(G) \geq d$ as the edges adjacent to any single vertex must receive different colours. From Vizing's theorem we know that $\chi'(G) \leq d + 1$. We show that since $G$ has a cut vertex $v$, we can't have $\chi'(G) = d$, from which we will conclude that $\chi'(G) = d + 1$. Say we have a $d$-colouring $c$ of $G$. Let $C$ be a connected component of $G - v$. Let $vw$ be an edge from $v$ to a vertex $w$ of $C$ and let $vw'$ be an edge from $v$ to a vertex $w'$ outside $C$. The edges whose colour is equal to $c(vw)$ and are contained in $C$ form a perfect matching in $C - w$, which implies that $|C|$ is odd. The edges whose colour is equal to $c(vw')$ and are contained in $C$ form a perfect matching in $C$, which implies that $|C|$ is even. This is the required contradiction.

**Exercise 6** [10 points]

(1) Given $n$ men $m_1, \ldots, m_n$ and $n$ women $w_1, \ldots, w_n$, with a list $L(x)$ for each person $x$ that consists of a preference order for the $n$ members of the opposite gender, we want to find a stable perfect matching between the men and women. Here a stable matching is defined as a matching in which we do not have an unmatched pair $(m_i, w_j)$ such that $m_i$ prefers $w_j$ over his current partner and $w_j$ prefers $m_i$ over her current partner.

*Gale-Shapley Algorithm.*
**Input**: The preference lists of $n$ men and $n$ women.

**Output**: A stable matching.

**Iteration**: In the $i$-th round, each man proposes to the most preferred woman on his list, the women who are proposed to say 'maybe' to the best proposal she receives in this round and rejects everyone else. If there are no rejections in the round, then we terminate and return the current matching (with the 'maybe's'), otherwise the men crosses out the women who reject them and we go to the $(i+1)$-th round.

(2) Let $S$ be the stable matching returned by the proposal algorithm above, and suppose that it's not man-optimal. Let $i$ be the first round where some man proposes to a valid partner and gets rejected. Pick $m$ to be one such man in this round and let $w$ the valid partner it proposes to in round and gets rejected because of a man $m' \neq w$ who proposed to her and was higher on the preference list. Since $w$ is a valid partner of $m$, there exists a stable matching $S'$ in which $(m, w)$ are matched to each other. Let $w' \neq w$ be the partner of $m'$ in $S'$. We claim that $(m', w)$ is an unstable pair in $S'$. As we said above, $w$ prefers $m'$ to $m$. Since $m$' proposed to $w$, he must have been rejected by all of the women on his list before $w$. Since this is the first time a man is rejected by his best valid partner, $m'$ has not been rejected by his best valid partner in any rounds before $i$, and therefore by any of his valid partners. In particular, $w'$, which is a valid partner of $m'$, must be after $w$ on the list of $m'$. This shows that $(m', w)$ is an unstable pair in $S'$.