MTAT.03.286: Advanced Methods in Algorithms **University of Tartu**

Final Exam Solutions

Course staff: Vitaly Skachek, Pille Pullonen June 16th, 2017

Student name: ________________________________

Student ID: ________________________________

1. This exam contains 10 pages. Check that no pages are missing.
2. It is possible to collect up to 110 points. Try to collect as many points as possible.
3. Justify and prove all your answers (where applicable).
4. All facts and results that were proved or stated in the class can be used in your solution without a proof. Such results need to be rigorously formulated.
5. Any printed and written material is allowed in the class. No electronic devices are allowed.
6. Exam duration is 3 hours.
7. Good luck!

Question 1	
Question 2	
Question 3	
Question 4	
Total	

Question 1 (20 points).

By using Fast Fourier Transform (FFT) algorithm, evaluate the polynomial $P(x) = x^3 - x^2 + 5x - 3$ at the complex 4th roots of unity. Show at least one level of recursion.

We choose the 4th roots of unity so that $\omega^4 = 1$ and $\omega = i$. Hence $\omega^2 = -1$, $\omega^3 = -i$.

At first we write out the FFT matrix and then we rearrange the columns.

$$\begin{pmatrix} A(1) \\ A(\omega) \\ A(\omega^2) \\ A(\omega^3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & 1 & \omega^2 \\ 1 & \omega^3 & \omega^2 & \omega \end{pmatrix} \cdot \begin{pmatrix} -3 \\ 5 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega^2 & \omega & \omega^3 \\ 1 & 1 & \omega^2 & \omega^2 \\ 1 & \omega^2 & \omega^3 & \omega \end{pmatrix} \cdot \begin{pmatrix} -3 \\ -1 \\ 5 \\ 1 \end{pmatrix} =$$

$$= \begin{pmatrix} M_2(\omega^2) & E \cdot M_2(\omega^2) \\ M_2(\omega^2) & F \cdot M_2(\omega^2) \end{pmatrix} \cdot \begin{pmatrix} -3 \\ -1 \\ 5 \\ 1 \end{pmatrix}$$

Here

$$E = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \qquad F = \begin{pmatrix} \omega^2 & 0 \\ 0 & \omega^3 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -\omega \end{pmatrix} = -E$$

We now recursively evaluate

$$M_2(\omega^2) \cdot \begin{pmatrix} -3 \\ -1 \end{pmatrix} \qquad M_2(\omega^2) \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix}$$

We can write this out using FFT to reduce this to multiplying with $M_1 = 1$.

$$M_2(\omega^2) \cdot \begin{pmatrix} -3 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & \omega^2 \end{pmatrix} \cdot \begin{pmatrix} -3 \\ -1 \end{pmatrix} = \begin{pmatrix} M_1 & M_1 \\ M_1 & \omega^2 M_1 \end{pmatrix} \cdot \begin{pmatrix} -3 \\ -1 \end{pmatrix} = \begin{pmatrix} -4 \\ -2 \end{pmatrix}$$

$$M_2(\omega^2) \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & \omega^2 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} M_1 & M_1 \\ M_1 & \omega^2 M_1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

Now, we can put this together

$$\begin{pmatrix} A(1) \\ A(\omega) \\ A(\omega^2) \\ A(\omega^3) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} -4 \\ -2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 4 \end{pmatrix} \\ \begin{pmatrix} -4 \\ -2 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 4 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} -4+6 \\ -2+4\omega \\ -4-6 \\ -2-4\omega \end{pmatrix} = \begin{pmatrix} 2 \\ -2+4i \\ -10 \\ -2-4i \end{pmatrix}$$

Question 2 (25 points). Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a finite connected undirected graph with the real-valued weight function w defined on the edges, such that $w(e) > 0$ for all edges $e \in \mathcal{E}$. Assume that $\mathcal{G}'(\mathcal{V}', \mathcal{E}')$ is a *connected* subgraph of $\mathcal{G}$ (in particular, $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$).

Prove or disprove the following statements.

(a) If $\mathcal{T}(\mathcal{V}, \mathcal{E}_T)$ is a minimum spanning tree of $\mathcal{G}$ then the set of edges $\mathcal{E}_T \cap \mathcal{E}'$ defines a minimum spanning tree of $\mathcal{G}'$.

(b) If $\mathcal{T}'(\mathcal{V}', \mathcal{E}_S)$ is a minimum spanning tree of $\mathcal{G}'$, then there exists a spanning tree $\mathcal{T}(\mathcal{V}, \mathcal{E}_T)$ of $\mathcal{G}$ such that $\mathcal{E}_S \subseteq \mathcal{E}_T$ and $\mathcal{T}$ is a *minimum* spanning tree.

Both of the statements are actually false as we can show with simple counterexamples.

Counterexample for a) Consider a complete graph $\mathcal{G}$ with four vertices a, b, c, d. Assume that $w(\{a, b\}) = w(\{b, c\}) = w(\{c, d\}) = 1$, and $w(e) = 3$ for all other edges $e \in \mathcal{V}$.

The minimum spanning tree $\mathcal{T}$ consist of edges $\{a, b\}, \{b, c\}, \{c, d\}$. Define $\mathcal{G}'$ as $\mathcal{V}' = \{a, b, c\}$ and edges $\mathcal{E}' = \{\{a, b\}, \{a, c\}\}$. The the set $\mathcal{E}_T \cap \mathcal{E}' = \{\{a, b\}\}$ and the it is not a spanning tree of $\mathcal{G}'$.

Counterexample for b) Let $\mathcal{G}$ be $\mathcal{V} = \{a, b, c, d\}$ and edges $\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\}$. Let $w(\{d, a\}) = 3$ and for all other edges e let $w(e) = 1$. It is clear that MST of $\mathcal{G}$ has edges $\{a, b\}, \{b, c\}, \{c, d\}$ and no MST can contain the edge $\{d, a\}$.

Define $\mathcal{G}'$ as $\mathcal{V}' = \{a, d\}$ and $\mathcal{E}' = \{\{d, a\}\}$. Then this edge is also a MST of $\mathcal{G}'$. There is no MST of the original graph $\mathcal{G}$ that has this MST has a sub-tree.

Question 3 (30 points). A flow network *with finite capacities of vertices* is defined as a 5-tuple $\mathcal{N}(\mathcal{G}(\mathcal{V},\mathcal{E}), s, t, b, c)$, such that $\mathcal{N}(\mathcal{G}(\mathcal{V},\mathcal{E}), s, t, c)$ is a standard flow network. Additionally, $b : \mathcal{V} \to \mathbb{Q}^+$ is a vertex capacity function, such that any flow function $f : \mathcal{E} \to \mathbb{R}$ in $\mathcal{N}$ must satisfy

$$\sum_{e \in \text{in}(v)} f(e) \leq b(v)$$

for any vertex $v \in \mathcal{V} \setminus \{s, t\}$.

Propose an efficient algorithm which finds a maximum flow in a network with finite capacities of vertices with time complexity $O(|\mathcal{V}|^2|\mathcal{E}|)$ and prove its correctness (you can assume that for any vertex $v \in \mathcal{V}$ there is a path from s to v and a path from v to t in $\mathcal{G}$).

Sketch of the solution

Algorithm Construct a new (regular) flow network as follows. Replace each vertex $v \in \mathcal{V}$ with a pair of vertices v_{in} and v_{out} and add a directed edge (v_{in}, v_{out}) with capacity $b(v)$ (if $v = s$ or $v = t$ then the edge capacity is $+\infty$). For each edge (u, v) in the original network the new flow network has an edge (u_{out}, v_{in}) with the same capacity as the original edge. Run Dinitz algorithm on the new network. Translate the flow to a flow in the original network by using the flow on (u_{out}, v_{in}) on the edge (u, v).

Correctness

The flow is a legal flow in the original network $\mathcal{N}$. First, the flow in the edge (v_{in}, v_{out}) in the new network is at most $b(e)$. Hence, the incoming flow to v_{in} is at most $b(e)$ as the vertex rule is satisfied by the Dinitz algorithm. The incoming flow to v in the original network is exactly the same as the incoming flow to v_{in} in the second network. Hence, the algorithm satisfies the required property

$$\sum_{e \in \text{in}(v)} f(e) \leq b(v)$$

for any vertex $v \in \mathcal{V} \setminus \{s, t\}$. Additionally, it can be shown that the edge and vertex rules are also satisfied.

The flow is maximal in $\mathcal{N}$. Assume that there exists an augmenting path in the original network $\mathcal{N}$ that satisfies the vertex capacities. In this case, there exists analogous augmenting path also in the new network (and this path has the same available capacity). However, by definition, there is no augmenting paths in the network after Dinitz algorithm has terminated. Contradiction.

Complexity The constructed network has $2|V|$ vertices and $|V| + |E|$ edges. However, if there is a path from s to each v and to t from v, then $|V| < |E|$. Thus, Dinitz algorithm has complexity $O(4|V|^2 \cdot (|V| + |E|)) = O(|V|^2|E|)$ as required.

Question 4 (35 points).

Definitions:

1. *An independent set* in an undirected finite graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a subset of vertices $\mathcal{C} \subseteq \mathcal{V}$, such that no two vertices in $\mathcal{C}$ are adjacent.
2. Given an undirected finite graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, *a maximum independent set problem* is a problem of finding an independent set of maximum size.
3. *An extreme point solution* is a feasible solution that cannot be expressed as a convex combination of other feasible solutions.
4. *A half-integral solution* is a feasible solution in which each variable is either 0, 1 or $\frac{1}{2}$. *An integral solution* is a feasible solution in which each variable is either 0 or 1.

We formulate the following relaxation of an integer linear programming problem for a maximum independent set problem:

$$\begin{aligned} \textsf{maximize} \quad & \sum_{v\in\mathcal{V}} x_v \\ \textsf{subject to} \quad & \forall e = \{u,v\} \in \mathcal{E} \ : \ x_u + x_v \le 1 \\ & \forall v \in \mathcal{V} \ : \ x_v \ge 0 \end{aligned}$$

Here, for all $v \in \mathcal{V}$, we used indicator variables x_v, which are 1 if $v \in \mathcal{C}$, and 0 otherwise.

(a) Show that any feasible solution, which is not half-integral, is not an extreme point solution.

(b) By using the result in (a), show that if the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is bipartite then all extreme point solutions are integral.

(c) Assume that the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is bipartite. Propose a linear-programming based algorithm for solving a maximum independent set problem in $\mathcal{G}$. Justify your solution.

Half-integrality Assume, by contradiction, that there is an extreme point solution $\vec{x}$ that is not half-integral. Based on this solution, define two sets of vertices: $V_+ = \{v|\frac{1}{2} < x_v < 1\}$ and $V_- = \{0 < x_v < \frac{1}{2}\}$.

Define two new solutions $\vec{t}$ and $\vec{w}$:

$$t_i = \begin{cases} x_i & \text{if } x_i \text{ is half-integral} \\ x_i + \varepsilon & \text{if } i \in V_+ \\ x_i - \varepsilon & \text{if } i \in V_- \end{cases} \qquad w_i = \begin{cases} x_i & \text{if } x_i \text{ is half-integral} \\ x_i + \varepsilon & \text{if } i \in V_- \\ x_i - \varepsilon & \text{if } i \in V_+ \end{cases}$$

It is clear that $\vec{x} = \frac{1}{2}\vec{t} + \frac{1}{2}\vec{w}$ so if $\vec{t}$ and $\vec{w}$ are feasible solutions then x is not an extreme point solution as it can be represented as a convex combination of two or more feasible solutions. Also, $\vec{x}$ is not half-integral, hence $\vec{w} \neq \vec{t}$.

It remains to show that for a proper choice of ε, $\vec{w}$ and $\vec{t}$ are indeed feasible solutions. We can ensure $\vec{v} > 0$ and $\vec{t} > 0$ by choosing ε small enough such that $x_i - \varepsilon > 0$ for each $x_i \neq 0$.

We also need to consider the edge constraints $x_u + x_v \leq 1$:

- $x_u + x_v < 1$ we can ensure this by small enough ε such that $x_u + x_v + 2\varepsilon \leq 1$ for each u, v as $x_u \neq 1$ and $x_v \neq 1$.
- $x_u + x_v = 1$
  - $x_u = 0$ and $x_v = 1$ (or vice versa) we have $x_u = t_u = w_u$ and $x_v = t_v = w_v$ so the constraint is satisfied.
  - $x_u = \frac{1}{2}$ and $x_v = \frac{1}{2}$ we have $x_u = t_u = w_u$ and $x_v = t_v = w_v$ so the constraint is satisfied.
  - $u \in V_+$ and $v \in V_-$ (or vice versa, in which case the claim is analogous) we have ε cancelling out in the constrains and $x_u + x_v = t_u + t_v = w_u + w_v$

Hence, the defined points are both feasible solutions and therefore $\vec{x}$ is not an extreme point solution.

Integrality in a bipartite graph From the previous part we already know that the extreme point solution is half-integral. Next, we show that only integral values can be extreme point solutions in the bipartite case. Let $\mathcal{V} = A \cup B$ be a partition of vertices of $\mathcal{G}$, such that all edges in $\mathcal{E}$ have one endpoint in A and one endpoint in B. Let $\vec{x}$ be the half-integral solution to the linear program. Define two new solutions. For all $i \in \mathcal{V}$:

$$t_i = \begin{cases} x_i & \text{if } x_i \text{ is integral} \\ 0 & \text{if } x_i = \frac{1}{2} \text{ and } i \in B \\ 1 & \text{if } x_i = \frac{1}{2} \text{ and } i \in A \end{cases} \qquad \text{and} \qquad w_i = \begin{cases} x_i & \text{if } x_i \text{ is integral} \\ 0 & \text{if } x_i = \frac{1}{2} \text{ and } i \in A \\ 1 & \text{if } x_i = \frac{1}{2} \text{ and } i \in B \end{cases}$$

It is clear that $\vec{x} = \frac{1}{2}\vec{t} + \frac{1}{2}\vec{w}$ so if $\vec{t}$ and $\vec{w}$ are feasible solutions then $\vec{x}$ is not an extreme point solution as it can be represented as a convex combination of two or more feasible solutions. Also, $\vec{x}$ is not integral, hence $\vec{w} \neq \vec{t}$.

It is clear that both $\vec{t}$ and $\vec{w}$ are non-negative. We also need to check that all constraints are satisfied for $\vec{t}$ and $\vec{w}$.

Consider an edge $\{u, v\}$ such that $u \in A$ and $v \in B$. We have the following possibilities:

- $x_u = 0$ and $x_v = 1$ (or vice versa) in which case $t_u = w_u = x_u$ and $t_v = w_v = x_v$ and the constraint is satisfied as it is satisfied for the solution $\vec{x}$.
- $x_u = \frac{1}{2}$ and $x_v = \frac{1}{2}$. In this case $t_u = 1$, $t_v = 0$, and $w_u = 0$, $w_v = 1$, and the constraints are satisfied.
- $x_u = \frac{1}{2}$ and $x_v = 0$. In this case $t_u = 1$, $t_v = 0$, and $w_u = 0$, $w_v = 0$, and the constraints are satisfied.
- $x_u = 0$ and $x_v = \frac{1}{2}$. In this case $t_u = 0$, $t_v = 0$, and $w_u = 0$, $w_v = 1$, and the constraints are satisfied.

The algorithm Since each extreme point solution is integral then we can use the fact that Simplex algorithm will find an extreme point solution to any linear program. Since the integral solution has a one to one correspondence with the solution based on the values of the indicator variables, we can just solve the linear program and directly define the independent set:

$$\mathcal{C} = \left\{ v \in \mathcal{V} \ : \ x_v = 1 \right\}$$