AI2615: Algorithm Design and Analysis
Final Examination

1. (25 points) In this question, we consider a variant of the network flow problem, where capacity constraints are placed on *vertices* instead of edges. Given a simple[1] directed graph $G = (V, E)$, a source $s \in V$ (you can assume the in-degree of s is 0), a sink $t \in V$ (you can assume the out-degree of t is 0), and a capacity function $c : V \setminus \{s, t\} \to \mathbb{R}_{>0}$, a *flow* is a function $f : E \to \mathbb{R}_{\geq 0}$ such that it satisfies the following two constraints:

   1. Flow conservation: for each $u \in V \setminus \{s, t\}$, we have
   $$\sum_{v:(v,u)\in E} f(v, u) = \sum_{w:(u,w)\in E} f(u, w);$$
   2. Capacity constraint: for each $u \in V \setminus \{s, t\}$, we have
   $$\sum_{v:(v,u)\in E} f(v, u) \leq c(u).$$

   The *value* of a flow f is defined in the same way as it is in the conventional network flow problem:
   $$\text{value}(f) = \sum_{u:(s,u)\in E} f(s, u).$$

   (a) (10 points) Design a polynomial-time algorithm for this variant of the max-flow problem.

   (b) (15 points) Given a directed unweighted simple graph $G = (V, E)$, a source $s \in V$ (you can assume the in-degree of s is 0) and a sink $t \in V$ (you can assume the out-degree of t is 0), a *vertex separator* is a subset of vertices $S \subseteq V \setminus \{s, t\}$ such that there is no s-t path after removing S. Design a polynomial-time algorithm for finding a vertex separator with minimum number of vertices.

   For both parts above, please prove the correctness of your algorithms and analyze their time complexities.

[1] A graph is simple if it contains no multiple edges and no self-loops. Notice that all the graphs we have seen in this course are simple graphs.

2. (25 points) Consider the following game with two players A and B. A *binary* string is given to the two players. The two players move alternatively. When a player moves, (s)he has two actions:

   1. remove the digit on the left end of the string;
   2. remove the digit on the right end of the string.

   The game ends when the last remaining digit in the string is removed. Suppose A moves first. The winner of the game is defined as follows:

   - If the last digit is removed by A, A wins if the digit removed is 1, and B wins if the digit removed is 0;
   - If the last digit is removed by B, B wins if the digit removed is 1, and A wins if the digit removed is 0.

   Design a polynomial-time algorithm to decide whether A has a strategy to guarantee that (s)he always wins (regardless of what B plays in each round). Prove the correctness of your algorithm and analyze its running time. Your algorithm only needs to *decide* whether A has a winning strategy; your algorithm does not need to find a winning strategy should it exist.

3. (25 points) Prove that the following two problems are NP-complete.

   (a) (15 points) Given an undirected and unweighted simple graph $G = (V, E)$ and two positive integers k and ℓ, decide if we can remove k vertices from G such that G contains at least ℓ isolated vertices. A vertex is isolated if its degree is 0.

   (b) (10 points) Given an undirected and unweighted simple graph $G = (V, E)$ and a positive integer k, decide if we can remove k vertices from G such that G contains at least k isolated vertices. In this problem, you can assume the decision problem in (a) is NP-complete even if you are not able to prove it.

4. (25 points) Consider the following algorithm which decides if an undirected and unweighted simple graph $G = (V, E)$ contains a vertex cover of size at most k.

Algorithm 1 Decide if G has a vertex cover of size at most k

$\textsc{Find}(G = (V, E), k)$

1: if G contains no edge, return **true**
2: if $k = 0$, return **false**
3: pick an arbitrary edge (u, v)
4: let $N(u)$ and $N(v)$ be the sets of edges incident to u and v respectively
5: return $(\textsc{Find}(G = (V \setminus \{u\}, E \setminus N(u)), k - 1) \vee \textsc{Find}(G = (V \setminus \{v\}, E \setminus N(v)), k - 1))$

(a) (10 points) Prove the correctness of this algorithm.

(b) (15 points) Analyze its running time. The running time must be in terms of k and the size of the graph ($|V|$ or/and $|E|$). Notice that the running time may depend on the data structure you use to store the graph. Please specify the data structure you use.