## Problem Set No. 6

1. (CLR 27.2-8, p. 600) Show that a maximum flow in a network $G = (V, E)$ can always be found by a sequence of at most $|E|$ augmenting paths. (*Hint:* Determine the paths *after* finding the maximum flow.)

2. (CLR 27.3-5, p. 604) A bipartite graph $G = (V, E)$ where $V = L \cup R$, is **d-regular** if every vertex $v \in V$ has degree exactly $d$. Every $d$-regular bipartite graph has $|L| = |R|$. Prove that every $d$-regular bipartite graph has a matching of cardinality $|L|$ by arguing that a minimum cut of the corresponding flow network has capacity $|L|$.

3. (CLR 36.1-6, p. 924) Show that an algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

4. (CLR 36.4-6, p. 946) Suppose that someone gives you a polynomial-time algorithm to decide formula satisfiability. Describe how to use this algorithm to find satisfying assignments in polynomial time.

5. (CLR 36-1, p. 961) ***Independent set***

   An **Independent set** of a graph $G - (V, E)$ is a subset $V' \subseteq V$ of vertices such that each edge in $E$ is incident on at most one vertex in $V'$. The **independent set problem** is to find a maximum-size independent set in $G$.

   ***a.*** Formulate a related decision problem for the independent-set problem, and prove that it is NP-complete. (*Hint:* Reduce from the clique problem.)

   ***b.*** Suppose that you are given a subroutine to solve the decision problem you defined in part (a). Give an algorithm to find an independent set of maximum size. The running time of your algorithm should be polynomial in $|V|$ and $|E|$, where queries to the black box are counted as a single step.

   Although the independent-set decision problem is NP-complete, certain special cases are polynomial-time solvable.

   ***c.*** Give an efficent algorithm to solve the independent-set problem where each vertex in $G$ has degree 2. Analyze the running time, and prove that your algorithm works.

***d.*** Give an efficient algorithm to solve the independent-set problem when $G$ is bipartite. Analyze the running time, and prove that your algorithm works correctly. (*Hint:* Use the results of Section 27.3.)

6. (CLR 37-1, p. 968) From the proof of Theorem 35.12, we know that the vertex-cover problem and the NP-complete clique problem are complementary in the sense that an optimal vertex cover is the complement of a maximum-size clique in the complement graph. Does this relationship imply that there is an approximation algorithm with constant ratio bound for the clique problem? Justify your answer.

7. (CLR 37-1, p. 983) ***Bin packing***

Suppose that we are given a set of $n$ objects, where the size $s_i$ of the $i$th object satisfies $0 < s_i < 1$. We wish to pack all the objects into the minimum number of unit-size bins. Each bin can hold any subset of the objects whose total size does not exceed 1.

***a.*** Prove that the problem of determining the minimum number of bins required is NP-hard. (*Hint:* Reduce from the subset-sum problem.)

The ***first-fit*** heuristic takes each object in turn and places it into the first bin that can accommodate it. Let $S = \sum_{i=1}^{n} s_i$.

***b.*** Argue that the optimal number of bins required is at least $\lceil S \rceil$.

***c.*** Argue that the first-fit heuristic leaves at most one bin less than half full.

***d.*** Prove that the number of bins used by the first-fit heuristic is never more than $\lceil 2S \rceil$.

***e.*** Prove a ratio bound of 2 for the first-fit geuristic.

***f.*** Give an efficient implementation of the first-fit heuristic, and analyze its running time.