BU CLA CS 530: Analysis of Algorithms

Spring 1999

Problem Set 12

Due Thursday April 15, by 5:00 pm, in my mailbox in the main office of the CS department. Late submissions not accepted.

---

1. (Review exercise.) Solve the following recurrence exactly, i.e., do not write a solution using "$\mathcal{O}$" or "Θ" or "Ω":

   $$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 3T(n/2) + n^2 - n & \text{if } n \geqslant 2 \text{ is a power of 2.} \end{cases}$$

   Hint 1: Use the iteration method (Section 4.2 in [CLR]).
   Hint 2: If you solve it my way, you will need to use the equation $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$. If you use this equation, you will also have to prove it!

2. Write the pseudo-code of an algorithm which, given a directed graph $G = (V, E)$, prints all Hamiltonian cycles in G. If G has n vertices, say $V = \{1, 2, \ldots, n\}$, your algorithm must run in time $\mathcal{O}(n!)$. (You will not get credit if you exceed this run time.)

   Assume the existence of an algorithm that generates, one by one, all the permutations of $\{1, 2, \ldots, n\}$ in time $\mathcal{O}(n!)$, i.e., in time linear in the total number of permutations.

3. Consider an input instance $S = \{s_1, \ldots, s_n\}$ of the PARTITION decision problem, where every s_i is a positive integer.[1] Show that if there is a polynomial function $f(n)$ such that $s_i \leqslant f(n)$ for every $i \in \{1, \ldots, n\}$, then this decision problem can be solved in polynomial time.

4. Consider the following decision problem, called VERTEX-DISJOINT-CONNECTIVITY: An input instance consists of an undirected graph G together with $k \geqslant 1$ disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The requested output is *Yes* or *No*, depending on whether or not there are k mutually vertex-disjoint paths, one from s_i to t_i for every $i \in \{1, \ldots, k\}$.

   Show that VERTEX-DISJOINT-CONNECTIVITY is NP-complete.
   Hint: Use a reduction from 3-CNF-SAT. Let k be the number of clauses plus the number of variables.
   Comment 1: For a fixed $k \geqslant 2$, the problem is probably solvable in polynomial time (I recall reading it somewhere, but was not able to locate a reference for it). For $k = 1$, the problem is clearly solvable "quickly", because it is just a matter of deciding whether there is a path from s_1 to t_1.
   Comment 2: An interesting variation is to include in the input k additional parameters $N_1, \ldots, N_k$ which are all positive integers and to ask whether or not there are N_i mutually

---

[1] The PARTITION problem is defined in Exercise 36.5-4, page 960, in [CLR].

vertex-disjoint paths from s_i to t_i for all $i \in \{1, \dots, k\}$. This is again NP-complete. For a fixed $k \geqslant 2$, the problem is probably polynomial-time solvable. And for $k = 1$, it is straightforward to reduce it to a maximum-flow problem, which is polynomial-time solvable by the techniques of Chapter 27 in [CLR].

Created: 99.04.08
Modified: 99.04.27