Exercises Algorithms and Networks – Set 7 - 2012 /2013

Deadline: Monday, November 5, 2012.

In case you need to email about this exercise set, please put [AN7] in the header.

The bonus question can give you extra points.

1. An exact algorithm for Capacitated Bipartite Dominating Set (2 points)

In this problem, we are given a bipartite graph $G = (S \cup C, E)$. S is the set of *servers*, C is the set of clients. Edges have one endpoint in S and one endpoint in C. $S \cap C = \emptyset$.

Each server has a capacity, i.e., we have a function $d : S \to \mathbf{Z}+$. So, capacities are positive integers.

The capacity of a server is the number of clients it can serve.

For each server, we can either *use* it, or *not use it.* If we use a server v, it can serve at most $d(v)$ clients, and only clients it has an edge to. If we do not use a server, it serves no clients, but also does not cost anything.

We consider the following problem:

Given is G with the function d, and an integer K. Can we use at most K servers, and then assign each client to a server we use, such that no server v has more than $d(v)$ clients assigned to it?

Give an algorithm that solves this problem in $O^*(2^n)$ time.

Hint: use a result that was shown in the earlier part of the course. You may also want to consult the slides of the first lecture in the course.

2. Dynamic Programming (2.5 points

Consider the following problem. Given is a graph $G = (V, E)$. How many Hamiltonian Circuits does G contain?

There is a detail that should be made clear with this question: when are circuits different? Here, we fix a vertex v, and look to circuits that start and end at v, but count circuits that are the same except that they go in the opposite direction as different circuits.

(a) Fix a vertex v. Argue that the total number of circuits equals the sum over all edges $\{v, w\} \in E$ of the number of Hamiltonian paths that start in v and end in w.

(b) Let v again be the fixed vertex. For a set of vertices $W \subseteq V$ with $v \in W$, and a vertex $x \in V$, let $N(W, x)$ be the number of paths that

- start in v
- end in x
- visit each vertex in W exactly one time
- do not visit vertices outside W

Give a recursive formulation of $N(W, x)$. (Hint: look at the Held-Karp algorithm.)

(c) Show that for each vertex $w \in V$, $N(V, w)$ can be computed in $O^*(2^n)$ time.

(d) Show that the number of Hamiltonian circuits of a graph can be computed in $O^*(2^n)$ time.

3. Analysing an Independent Set algorithm with a non-standard measure (5.5 points)

In this exercise, we will do a special case of an analysis with *a non-standard measure*. The method is a (somewhat simplified) form of the *Measure and Conquer* technique, see

> Fedor V. Fomin, Fabrizio Grandoni, Dieter Kratsch: A measure & conquer approach for the analysis of exact algorithms. J. ACM 56(5): (2009)

We will assume that G is a graph such that all vertices have degree at most three.

Consider the following algorithm for Independent Set.

(i) Explain how we can compute in polynomial time the maximum size of an independent set of a graph where all vertices have degree two.

(ii) Explain why the algorithm given on the next page (mis) is correct.

(iii) What can we say about the degree of the vertex x which is chosen before branching?

In order to analyze the running time of the algorithm better, we will use a nonstandard measure. Suppose we give every vertex of degree 0 or 1 a weight 0, every vertex of degree two a weight $\frac{1}{2}$, and every vertex of weight three a weight 1.

The *measure* of a graph G is the total weight of all vertices in G. We denote this by $m(G)$.

(iii) Consider the graph G^1 from the algorithm, i.e., the graph obtained by removing the vertex x in the branching step. Argue that $m(G^1) \leq m(G) - \frac{5}{2}$. (Hint: look what happens to the weights of the neighbors of v.)

Algorithm 1 mis(G)

Input: A graph $G = (V, E)$ with maximum degree three
Output: The maximum size of an independent set in G
if there exists a vertex $v \in V$ with $d(v) = 0$ **then**
  Return($1 + \text{mis}(G - \{v\})$;
end if
if there exists a vertex $w \in V$ with $d(w) = 1$ **then**
  Let G' be obtained by removing w and its neighbor.
  Return($1 + \text{mis}(G')$)
end if
if all vertices in G have degree two **then**
  Compute the size of a maximum independent set of G in polynomial time, say this value is α
  Return(α)
end if
Choose a vertex $x \in V$ of maximum degree.
Let G^1 be obtained from G by removing x.
Let G^2 be obtained from G by removing x and all neighbors of x.
Return($\max\{\text{mis}(G^1),\text{mis}(G^2) + 1\}$)

(iv) Consider the graph G^2 from the algorithm, i.e., the graph obtained by removing the vertex x and its neighbors in the branching step. Argue that $m(G^2) \leq m(G) - \frac{5}{2}$.

(v) Prove that the algorithm uses $O^*(c^n)$ time with c the positive solution of the equation

$$c^{\frac{5}{2}} - 2 = 0$$

(vi) Give the upper bound on the running time of the algorithm that follows from (v).

(vii) (Harder question.) Improve upon the claimed result of step (iv) by a better case analysis, and looking at vertices at distance two from G. (It is easy to make a mistake here; note that the neighbors of v can have edges to each other . . .)

(viii) Use your improved result from (vii) and improve upon the upper bound you obtained in (vi).

Bonus: can you get an even better bound by using a different measure? (E.g., not using $\frac{1}{2}$ but another value?)