McGill University COMP360 Winter 2011 **Instructor: Phuong Nguyen**

Assignment 7

Due March 9 before the lecture

The work you submit must be your own. You may discuss problems with each others; however, you should prepare written solutions alone. Copying assignments is a serious academic offense, and will be dealt with accordingly.

Question 1 (10pt) Consider the following variant of the Independent Set problem, called 3GIND here (G for generalized, and 3 for the maximum degree, see below). In this problem we are given an undirected graph G where the degree of each vertex is at most 3. Also, each vertex v in G is associated with a positive weight $w(v)$. The problem is to find an independent set of G of maximum total weight.

Consider the following greedy algorithm:

1. Let V be the set of all vertices of G, and S be the empty set $\varnothing$
2. while V is not empty do
3. pick a vertex v in V of maximum weight
4. add v to S
5. delete v and all its neighbors from V
6. end while
7. return S

(a)[1pt] Let S be the output of the algorithm. Show that S is an independent set.

(b)[4pt] Let T be any independent set of G. Show that for each vertex v in T:

- either v is also in S, or
- there is a vertex u in S such that $w(v) \leq w(u)$ and (v, u) is an edge of G

(c)[5pt] Show that the algorithm returns an independent set of total weight at least $1/3$ times the maximum total weight of any independent set in G.

Question 2 (10pt) Recall that Vertex Cover is the following problem:
Input: An undirected graph G and a positive integer k.
Output: Accept if and only if G has a vertex cover of size at most k.
(A vertex cover of G is a set of vertices that contains at least one endpoint of every edge in G.)

This problem is **NP**-complete, and so is MINVC, the problem of computing the minimum size of a vertex cover. Formally, the problem MINVC is specified as follows:
Input: An undirected graph G
Output: A minimum-size vertex cover of G.

Now consider the following greedy algorithm for approximating MINVC:
On input G:

1. Let S be the empty set $\varnothing$,
2. while G contains some edge do
3. let v be the vertex in G of maximum degree
4. add v to S
5. remove v and all edges incident on it from G
6. end while
7. output S

(a)[1pt] Let S be the output of the algorithm. Show that S is a vertex cover of the given graph G.

(b)[8pt] Show that S has size at most $c \log n$ times the minimum size of a vertex cover of G for some positive constant c, where n is the number of vertices of G.

(c)[1pt] According to your argument in (b), what is the constant c?