

Homework 7

22C:44 Algorithms
Due Tuesday, April 30, 2002

1. Solve Problem 16.2-4 (page 384) in the textbook. After specifying your algorithm, explain why your algorithm is “greedy algorithm.” You are *not* required to prove this algorithm correct.
2. Solve Problem 16.2-5 (page 384) in the textbook. A very simple greedy algorithm is sufficient for solving this problem. So the emphasis in this problem is the proof that your algorithm is correct. I want you to be extremely careful in making sure that you have proof that is clear and precise.
3. Given a graph $G = (V, E)$, a *vertex cover* of G is a subset V' of vertices such that for every edge $\{u, v\}$ in G , either $u \in V'$ or $v \in V'$ (or both). A *minimum vertex cover* of G is a vertex cover of G with fewest vertices. Consider the following greedy algorithm for computing a minimum vertex cover of a graph.
 - (a) $C := \emptyset$
 - (b) Pick a vertex v with maximum degree and add it to set C .
 - (c) Delete v from G (along with all edges incident on v).
 - (d) Stop, if G is empty. Otherwise goto Step (b).

Show a tree with 10 vertices for which this algorithm does not produce a minimum vertex cover.

Notes: (i) The *degree* of a vertex is the number of edges incident on it. (ii) If there are several vertices with maximum degree, the algorithm picks any one of these, arbitrarily.

4. In the programming assignment (Homework 10), we defined a graph G_n whose vertices are permutations.
 - (a) Draw the graph P_3 with the vertices labeled clearly with 3-permutations.
 - (b) Show the result of executing breadth first search on this graph starting with vertex $(1, 2, 3)$. In particular, show the breadth first search tree and the values of $d[v]$ for each vertex v .
 - (c) Show the result of executing depth first search on this graph starting with vertex $(1, 2, 3)$. Assume that in the adjacency list representation of this graph, the neighbors of each vertex are sorted in lexicographic order. Show the depth first search tree and list the back edges in the graph.