College of Computer & Information Science
Northeastern University
CSG399: Approximation Algorithms

Spring 2004
Handout 1
14 January 2003

Problem Set 1

1. Problem 34-3, page 1019 of [CLRS] (Problem 36-2 of [CLR]).

2. Exercise 1.4, page 8.

3. Exercise 1.7, page 8.

4. Give proofs of self-reducibility for the maximum matching and shortest superstring (Problem 2.9 of text) problems. That is, for each problem, show that given an oracle for the decision version of the problem, one can compute an exact solution to the optimization version in polynomial time. See Exercise 1.14 of text.

5. Edge Cover

An *edge cover* of a graph $G = (V, E)$ is a subset C of E such that for every vertex $u \in V$, there exists at least one edge in E that is incident on u. This exercise concerns the problem of finding a minimum-size edge cover of G.

- **(a)** Given a matching M of G, show how to construct an edge cover of G of size at most $|V| - |M|$.
- **(b)** Given an edge cover C of G, show that there exists a matching of G of size at least $|V| - |C|$.
- **(c)** Using parts (a) and (b) argue that a minimum edge-cover of G can be constructed efficiently from a maximum matching of G. Since the maximum matching of a graph can be constructed in polynomial time, this implies that the minimum edge cover problem is also solvable in polynomial time.
- **(d)** Consider the following greedy algorithm for the minimum edge cover problem. Set $G' = (V', E')$ to $G = (V, E)$ and C to $\emptyset$. Repeat the following steps (i), (ii), and (iii) until G' is empty: (i) Pick any edge (u, v) in G'. (ii) Add (u, v) to C. (iii) Remove vertices u and v from V' and remove any edges adjacent to u or v from E'. Finally, when the preceding repeat-loop terminates, for every vertex u not covered by an edge in C, add any edge (u, v) in E to C. Return C.

  Prove that the above greedy algorithm yields a 1.5-approximation for the minimum edge cover problem.

6. Exercise 2.2, page 23.