CS570

Analysis of Algorithms

Fall 2004

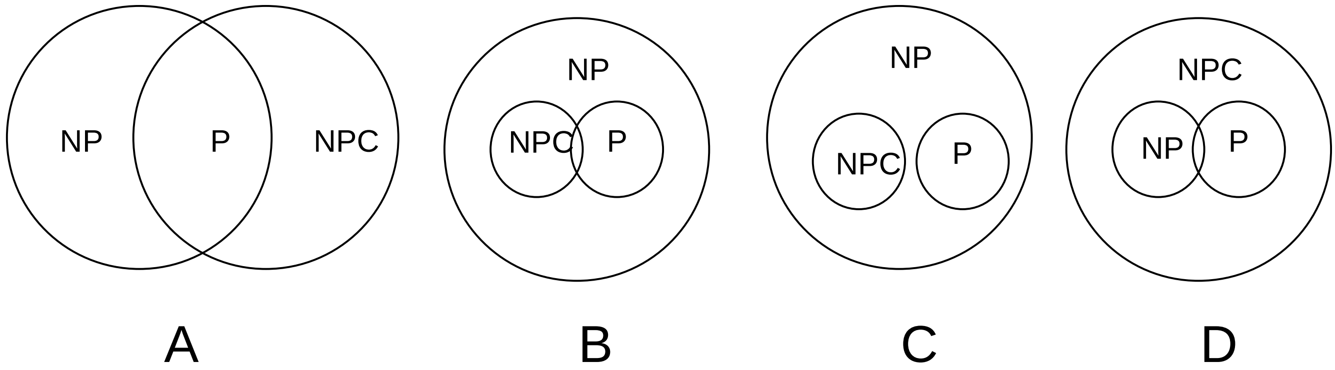
Final Exam

Name: ______________________

Student ID: __________________

1)

a. Which of the diagrams below best describes the relationship between P, NP, and NP-Complete problems? (25 pts)



b. Each diagram represents 3 relationships between P & NP, P & NPC, and NP & NPC. Prove each of the three relationships or state that a proof does not exist. If a proof does not exist state why this relationship should exist.

P & NP:

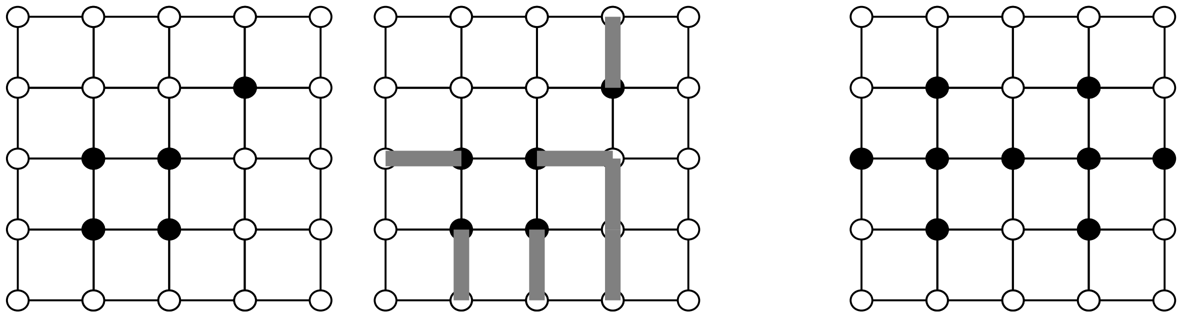
P & NPC:

NP & NPC:

2) Let $G = (V,E)$ be a directed graph with edge weights $w(e) > 0$ for all edges. For all paths p, let $m(p) = \max\{w(e)$, such that e is an edge on the path p$\}$. Let the set of vertices, $V = \{1, ..., n\}$. For all pairs $(i, j) \in V$, let $d(i, j) = \min(\{m(p)$, such that p is a path from i to j$\}$ or $d(i, j)=\infty$ if no such path exists. Show that $d(i, j)$ for all i, j with $1 \leq i \leq n$ and $1 \leq j \leq n$ can be computed in $O(n^3)$ time (in other words present the algorithm and proof). (25 pts)

Hint: First show that for all i, j, k, $d(i, j) \leq \max\{d(i, k), d(k, j)\}$.

3) Consider an n x n two-dimensional grid of nodes, with edges forming the grid lines. Thus, each node except for the edge nodes has four adjacent nodes. Some nodes are designated as “starting" nodes. The problem is to find a set of non-intersecting paths from the starting nodes to the edge of the grid, if such a set exists (the lengths of the paths do not matter. Non-intersecting paths cannot share edges or nodes. Use a network approach to model this problem. Describe your approach and give its worst-case running time in terms of n. (Hint: How can you prevent paths from intersecting at nodes?) (25 pts)



Examples: The black circles represent starting nodes. The diagram on the left has one solution given in the middle diagram (there are other solutions). Note that paths may turn as many corners as desired. The diagram on the right has no feasible solution.

4) Consider the following variation of the load balancing problem where we are given m machines with the property that the fastest machine is at most twice as fast as the slowest machine. We are given n jobs; each job has a processing time t_j. We seek to assign each job to one of the machines so that the loads placed on all machines are as "balanced" as possible.

For this heterogeneous environment devise a greedy approximation algorithm that produces a solution with no more than 3 times the value of the optimal solution. In specific, the quantity to be optimized is the maximum load on any machine (Maximum T_i , or maximum total load on any machine $1 \leq i \leq m$) Prove the bound on your algorithm. (25 pts)