EECS 477. HOMEWORK 6.

Due on Thursday 12/5/2002 before noon
in mailbox labeled 477 in room 2420 EECS

YOUR NAME: ____

You must show all work to receive credit!

Please read the statement below and sign your name; otherwise, your homework will not be graded. The text of the College of Engineering's Honor Code can be found at http://honor.personal.engin.umich.edu/

I hereby acknowledge that I understand the College of Engineering's Honor Code and have pledged to uphold it and abide by it.

Signature: ____

1. Decision problems(20pts)

A. State the decision version of the knapsack problem: given n items of known weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$ and a knapsack of capacity W, find the most valuable subset of the items that fits into the knapsack. The objects cannot be broken.

B. Outline a polynomial-time algorithm that verifies whether or not a proposed solution (certificate) solves the knapsack decision problem.

2. Paths(25pts)

Do problem 12.17(a) from the book

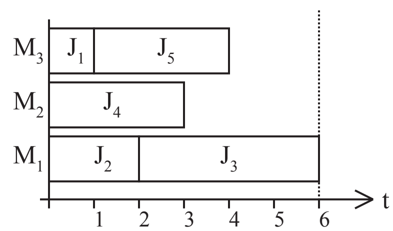
3. List scheduling

A.(25pts) Consider the following problem: we have n jobs, $J_1, \dots, J_n$, and m identical machines, $M_1, \dots, M_m$. Each job J_j must be processed without interruption for a time $p_j > 0$ on one of the m machines, each of which can process at most one job at a time. We need to find the schedule that minimizes the total makespan, that is, the time by which all jobs complete their processing.

Each schedule is specified by the sequences of jobs assigned to every machine in order they are processed. For instance, given three machines and five tasks with processing times

$$p_1 = 1, p_2 = 2, p_3 = 4, p_4 = 3, p_5 = 3.$$

Then one possible schedule will be specified as $[[J_2, J_3], [J_4], [J_1, J_5]]$. We can visualize this schedule with the following figure (the makespan is equal to 6 in this case):



The following **approximate** algorithm is proposed for this problem:

> the jobs are given in some particular order. Whenever a machine becomes available, the next job on the list is assigned to begin processing on that machine. If there is a tie, a machine with the smaller index takes the job.

Here is what you need to do: Show the schedule obtained by running the above algorithm on the instance with three machines ($m = 3$), and seven tasks $J_1, \dots, J_7$ with processing times given as

$$p_1 = 5, p_2 = 2, p_3 = 4, p_4 = 3, p_5 = 6, p_6 = 1, p_7 = 12.$$

Illustrate it graphically.

Now run the same algorithm on the sequence of jobs in inverse order, and make an illustration.

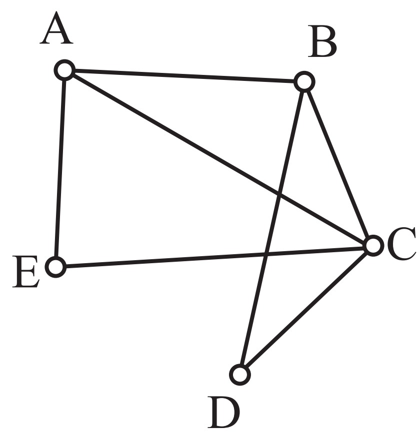
What is the ordering of the jobs for which the above algorithm gives the worst answer (that is, the longest makespan)?

What is the schedule that minimizes the makespan?

See if your answers satisfy the statement of the part B of this problem.

B.(extra 20 points) Prove that the above approximate algorithm always produces a solution whose makespan is at most twice the optimal one. That is, if the optimal makespan is T, the makespan obtained by the above algorithm is less than $2T$.

4. Graph problems (30 pts)



For the graph above,

A. Find all the hamiltonian cycles.

B. Find its chromaticity number (that is the minimal number of colors required to color the vertices of the graph so that no edge has the ends of the same color).

C. A *clique* is a set of nodes such that there is an edge in the graph between any two nodes in the clique. Find the size of the largest clique in the above graph.