CS 577: Introduction to Algorithms — Homework 4

Out: 10/06/10 — **Due: 10/13/10**

Ground Rules

- **Grading.** You will be graded on the correctness as well as clarity of your solutions. You are required to prove any claims that you make. In particular, when you are asked to design an algorithm, you must argue its correctness as well as running time. You may use without proof any theorems proven in class or in the textbook, as long as you state and cite them properly.
- **Collaboration.** You may work on and submit solutions in pairs.
- **Lateness.** Homework is due promptly at the start of class. Late homework will receive zero credit.
- **Extra credit questions.** Extra credit questions will not directly contribute to your score. However, they will be taken into account in the final grading and may improve your overall grade if your total score is close to the boundary between two grades. Furthermore they are fun to solve and improve your understanding of the course.
- Start working on your homework early. Plan your work in such a way that you have the opportunity to put some problems on the back burner for a while and revisit them later. Good luck!

Problems

1. **(4 points.)** Give an efficient algorithm for the following scheduling problem. You are given n jobs, each with a specific running time. Job i has a running time of t_i. You are required to schedule these on a machine one after another. Your goal is to minimize the total time that the jobs spend waiting, in particular,

$$\sum_{i=1}^{n} (\text{time at which job } i \text{ is completed})$$

   For example, if the jobs have running times of $1, 5$, and 4, and are scheduled in that order, the times at which they are completed are $1, 6$, and 10 respectively. So the total time that they spend waiting is $1 + 6 + 10 = 17$. On the other hand, if they are scheduled in the order of $4, 5, 1$, then the total time they spend waiting is $4 + 9 + 10 = 23$. Your goal is to find an ordering that minimizes this total time. Prove the correctness of your algorithm.

2. **(4 points.)** Problem 4.20 in the textbook (Pg. 199-200).

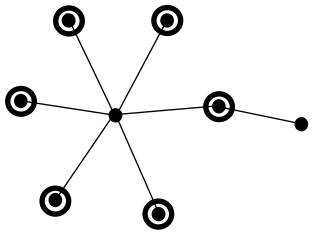
3. **(5+2 points.)** A *vertex cover* of a graph $G = (V, E)$ is a subset of V, V', such that *every* edge in E is incident on some vertex in V'. The first three figures on the next page show graphs with vertex covers—each vertex in the cover is circled. You should verify that each example gives a valid vertex cover.

   In the minimum vertex cover problem, our goal is to find a vertex cover of a given graph with the fewest number of nodes. Figures (i) and (ii) on the next page display two different vertex covers for the same graph; the one on the left (i) has more nodes than the one on the right (ii); the latter is the minimum vertex cover for the graph.

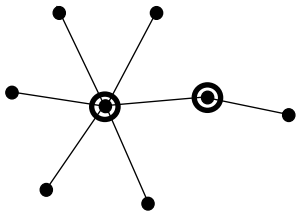
   (a) Find the minimum vertex cover for the trees in figures (iv), (v), (vi) and (vii) on the next page. (Feel free to mark your solution on that page and submit the page with the rest of your homework.)

   (b) Give a linear-time greedy algorithm that, given a tree, finds the minimum vertex cover in that tree. Prove the correctness of your algorithm. (A slightly worse running time, like $O(n \log n)$ or $O(n^2)$ will receive partial credit.)

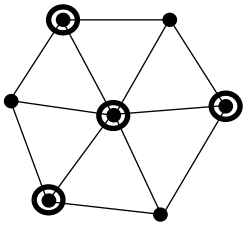
4. **(Extra credit.)** Trimedia Disks Inc. has developed "ternary" hard disks. Each cell on a disk can now store values 0, 1, or 2 (instead of just 0 or 1). To take advantage of this new technology, provide a modified Huffman algorithm for constructing an optimal variable-length prefix-free code for characters from an alphabet of size n, where the characters occur with known frequencies $f_1, f_2, \cdots, f_n$. Prove that your algorithm is correct.



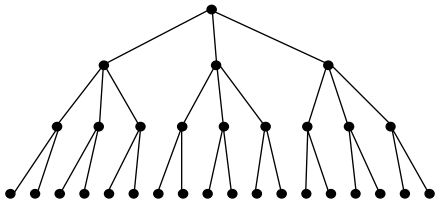
(i)



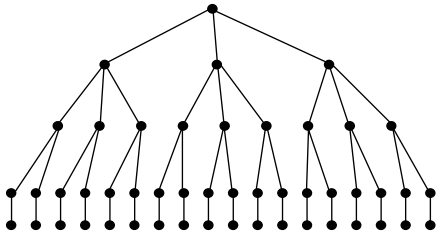
(ii)



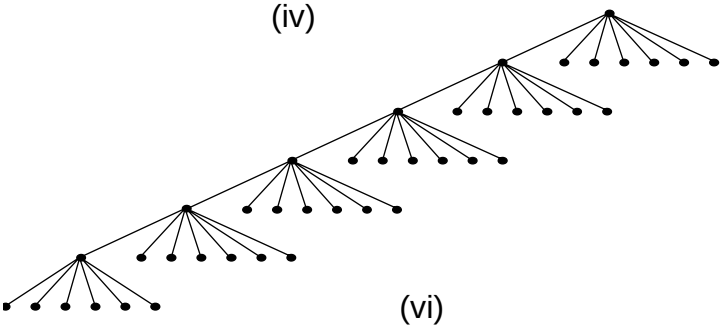
(iii)



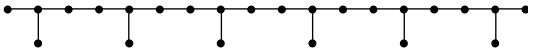
(iv)



(v)



(vi)



(vii)