CMPS 6610 Review/Homework Extra Credit

In this extra credit assignment, we will test and review concepts you have learned since the midterm exam. Please add your written answers to `answers.md` which you can convert to a PDF using `convert.sh`. Alternatively, you may scan and upload written answers to a file named `answers.pdf`.

1. **Algorithmic Paradigms**

What is your favorite algorithmic paradigm, and why?

2. **Divide and Conquer**

Do problems that can be solved by a divide and conquer approach necessarily satisfy the optimal substructure property? If so, provide a proof. If not, provide a counterexample.

3. **Randomization**

We learned in lecture that Quicksort takes $O(n \log n)$ expected work. For a random variable X, Markov's inequality states that:

$$\mathbf{P}[X \geq \alpha] \leq \frac{\mathbf{E}[X]}{\alpha}$$

3a). What is the probability that Quicksort does $\Omega(n^2)$ comparisons?

3b) What is the probability that Quicksort does $10^c n \ln n$ comparisons, for a given $c > 0$? What does this say about the "concentration" of the expected work for Quicksort?

4. **Greedy Algorithms**

Consider a scheduling problem where we are given n jobs $j_1, j_2, \ldots, j_n$, each of which has a processing time p_i. A schedule S is simply an ordering of jobs; each job will have a *waiting time* given by the sum of all processing times of jobs prior to it. Let us define the cost $C(S)$ of a schedule S to be the average waiting time over all jobs. Prove that scheduling jobs in order of their processing times (i.e., shortest-job-first) results in an optimal schedule.

5. **Dynamic Programming**

Problems that can be solved by dynamic programming require the optimal substructure property. We showed that the optimal substructure property induces a directed acyclic graph that can be used to derive the span of a dynamic programming algorithm. Give one example of an optimal substructure recurrence that has maximum span (i.e., no parallelism is possible) and one example that has polylogarithmic (i.e., ideal) span.

6. **Graphs**

You learned the cut property for minimum spanning trees. There is another useful fact called the *cycle property* for minimum spanning trees which states the following:

Given a graph G that contains a cycle, then the largest weight cycle in that graph **cannot** be contained in any minimum spanning tree.

Prove the cycle property.