ASSIGNMENT 1 CMSC 451 (Section 0101, Fall 2020)

Due by 11:59 pm on Wednesday, September 16. Submit your solutions in PDF via Gradescope.

1. *Asymptotic growth.* [10 points]

   List the following 5 functions in ascending order of asymptotic growth rate. (In other words, if $f(n)$ precedes $h(n)$ in your list, then it should be that $f = O(h)$.) The base of log is always 2.

   $$g_1(n) = 2^{\sqrt{\log n}}, \quad g_2(n) = n^{5/2}, \quad g_3(n) = 5^{\log n}, \quad g_4(n) = \log^7 n, \quad g_5(n) = n^6 + \log^2 n$$

2. *Gale-Shapley execution.* [25 points]

   Run the Gale-Shapley algorithm on the instance below, where each row indicates the ranked order of the indicated candidate/employer (highest ranked first).

Candidate preferences		Employer preferences	
candidate 1:	3, 7, 4, 2, 1, 5, 6	employer 1:	1, 6, 4, 3, 2, 5, 7
candidate 2:	4, 2, 7, 5, 1, 3, 6	employer 2:	4, 3, 7, 1, 6, 5, 2
candidate 3:	7, 2, 3, 4, 1, 6, 5	employer 3:	1, 6, 4, 2, 3, 5, 7
candidate 4:	4, 6, 2, 3, 1, 7, 5	employer 4:	7, 3, 6, 2, 4, 1, 5
candidate 5:	6, 7, 5, 2, 4, 3, 1	employer 5:	1, 6, 7, 2, 3, 5, 4
candidate 6:	5, 4, 1, 7, 3, 2, 6	employer 6:	4, 7, 6, 1, 5, 3, 2
candidate 7:	7, 6, 3, 4, 5, 1, 2	employer 7:	1, 5, 4, 3, 6, 7, 2

   List which employer each candidate is paired with. (*Note:* You should either run the algorithm by hand, or implement it in a programming language of your choice. You may *not* use someone else's implementation.)

3. *Stable matching for a class project.* [15 points]

   Suppose we want to assign partners in a class of $2n$ students. Each student provides a ranking of the other $2n - 1$ students; a *stable matching* is an assignment of partners so that no two students who are not partners would each prefer to work with each other rather than with their assigned partner. Show that a stable matching need not exist by giving an input instance (set of rankings) with $n = 2$ and proving that no stable matching is possible for that instance.

4. *Stable matching with incomplete rankings.*

   Consider a variant of the stable matching problem where a candidate need not rank every employer but can instead mark some employers as *excluded*. (If a candidate c excludes an employer e, then c prefers not to be matched at all rather than to be matched to e.) We call a matching *acceptable* if no candidate is matched with an employer that they exclude. Assume all candidates rank at least one employer.

   (a) [10 points] Show that a perfect, acceptable matching need not exist.

   (b) [20 points] Propose an algorithm that outputs a nonempty, stable, acceptable matching. (Stability is defined as usual; a candidate prefers any employer they ranked to an employer they excluded.) Give pseudocode for your algorithm along with a proof of correctness and an analysis of its running time.

5. *When are BFS and DFS the same?* [15 points]

   Fix a connected graph G and a specific vertex s in that graph. Suppose we obtain the same tree T when computing a breath-first search tree rooted at s and a depth-first search tree rooted at s. Prove that $G = T$. (*Hint:* you may use any results proved in Chapter 3 of the book, even if we did not cover them in class.)

6. *Collaboration.* [5 points]

   Write "I understand the course collaboration policy and have followed it when working on this assignment." List the other students with whom you discussed the problems, or else indicate that you did not discuss any problems with your classmates.