Homework 0

Due date: Solve, but do not turn in.

Purdue ID: id@purdue.edu

Date: January 29, 2019

I collaborated with (...). I affirm that I wrote the solutions in my own words and that I understand the solutions I am submitting.

Piazza

Sign up on Piazza (see link on class website). We ask that you use Piazza to discuss course content and to ask questions about homework problems. Asking good questions and contributing good answers on Piazza will count towards your course participation grade.

Question 2: Solve, but don't turn in (questions may appear in exams)

1. What is the running time of merge sort if the input array looks like

$$\langle 3, 2, 1, 6, 5, 4, 9, 8, 7, ..., 3n, 3n-1, 3n-2 \rangle?$$

   What is the running time of merge sort on an already sorted array? What is the runtime on an inversely sorted array?

2. What is the running time of bubble sort if the input array looks like

$$\langle 3, 2, 1, 6, 5, 4, 9, 8, 7, ..., 3n, 3n-1, 3n-2 \rangle?$$

   What is the running time of bubble sort on an already sorted array? What is the runtime on an inversely sorted array?

3. Rank the following functions by increasing order of growth (i.e., the slowest-growing first, the fastest-growing last):

   $n^{3/4}$, $n!$, $(\log n)^{400}$, $n^{3^{1000}}$, $n(\log n)^3$, $\binom{n}{n/3}$,$3n \log n$, 6^n, $n^{1.00001}$, $\log n^{30n}$, $(\log n)^{0.2}$

   where all the logarithms are to the base 2. If two functions have equal orders of growth then list them grouped together, e.g., between brackets {like this}.

4. Consider the recurrence $T(n) = T(\lceil n/6 \rceil) + 1$ and $T(x) = 1$ for $x \leq 6$. Solve this recursion by any method you know. For simplicity you may assume that n is a power of six so that $n/6$ is always an integer. Can you use induction to give an alternative proof of the solution you found in the previous step?

5. Bob picks a number y from 1 to $N = 2^n$, and asks Alice to guess it. The rule is that Alice can ask him questions about the number, but each question has a specific form. In particular, for each question Alice submits 5 numbers $1 \leq x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq N$ and Bob will respond with

   - 1 if $1 \leq y \leq x_1$
   - 2 if $x_1 < y \leq x_2$
   - 3 if $x_2 < y \leq x_3$
   - 4 if $x_3 < y \leq x_4$
   - 5 if $x_4 < y \leq x_5$
   - 6 if $x_5 < y \leq N$

   Describe a strategy that ensure that Alice finds out the number by asking at most $T(N)$ questions where $T(N)$ is defined as above.

6. Alice needs to climb n stairs to get to her room. In a single step Alice can climb 1 stair, 2 stairs or 3 stairs. Let $g(n)$ denote the total number of ways in which Alice can climb the stairs. Write down a recurrence relationship for $g(n)$. What is the number of ways in which Alice can climb the stairs if, say $n = 15$? Two walk sequences are considered different if the sequence of steps taken is not identical. (Hint: Try to develop a recursive formula for $g(n)$. Don't forget the base cases!).

7. Given an array of n integers and a target integer D, describe the best (in terms of running time) algorithm you can, that finds two elements of the array whose difference is D or outputs that no such elements exist.

8. Suppose s and t and u are vertices of an undirected graph G with n vertices and m edges. Suppose that the shortest path between s and t has k_t edges and the shortest path between s and u has k_u edges. Which of the following claims are necessarily true? If the claim is false provide a counter-example. Otherwise, provide a short proof.

   - The shortest path between t and u contains *at most* $k_t + k_u$ edges.
   - If $k_t < k_u$ then a Depth-First Search starting from node s will always node t before node u.
   - If $k_t < k_u$ then a Breadth-First Search starting from node s will always visit node t before node u.

9. Suppose s and t are vertices of an undirected graph G with n vertices and m edges. How fast can you determine if there is a path from s to t using *at most* k edges? What algorithm do you use?

10. Suppose that we want to find a stable matching for n men and n women. Prove or provide a counter-example to the following statements:

    (a) There is no stable matching in which every man is assigned to his least favorite partner.

(b) In any stable matching it must be the case that at least someone is matched with their favorite partner.

(c) If man m is first on woman w's preference list and woman w is first on man m's preference list then in *any* stable matching m will be paired with w.

(d) Suppose that M and M' are both stable matchings and that man m is matched with woman w in M, but with woman w' in M' where m m prefers w to w'. The Gale-Shapley algorithm will never output the matching M'.

11. Chapter 1. Problem 8. Kleinberg-Tardos (KT)