MIDTERM EXAM

CIS 5515 Design and Analysis of Algorithms (Spring 2021)

Note: For answer to each question, please explain your answer **in plain English** first. There are a total of 100 pts plus 10 bonus points. It is your responsibility to make sure that you have all the pages!

NAME:

(**Problem 1**. 20 points, 4 points each) **True** or **False**?
For each of the statements below, determine whether it is True or False. Briefly explain your answer or provide a counterexample for each of them.

(a) An $\Theta(n \log n)$ algorithm always runs faster than an $\Theta(n^2)$ algorithm.

False. $\Theta(n \log n)$ may have a large coefficient.

(b) Let G be an undirected graph on n nodes. G does not contain a cycle if (1) G is connected and (2) G has $n-1$ edges.

True. Since removing any edge will disconnect G, a cycle does not exist.

(c) The solution to the recurrence $T(n) = T(n/4) + T(3n/4) + cn$ is $T(n) = O(n \log n)$.

True. The cost at each level of the recursion tree is bounded by cn (e.g., $c(n/4)+c(3n/4) = cn$ at level 2 and $c(n/16)+c(3n/16)+c(3n/16)+c(9n/16) = cn$ at level 3). The depth is bounded $\log_{4/3} n$. Therefore, the total cost is $O(n \log n)$. Can also be proved by induction.

(d) The dynamic programming solution (Bellman-Ford) for the shortest path problem offers a solution for a more general setting than the greedy solution (Dijkstra's algorithm).

True, Dijkstra's algorithm cannot be used when there is a negative edge.

(e) There is a major difference between the way the binary search and the golden ratio search is conducted.

True, the binary search is used when the data is monotonically increasing or decreasing, while the golden ratio search is used when the data follows a quadratic function.

(**Problem 2**. 20 points) Suppose we are considering a stable matching between 3 boys (A, B, and C) and 3 girls (1, 2, 3). The boys' preference orders are the following: A: 213 (i.e., 2 is the most preferable and 3 is the least preferable by A), B: 132, C: 123, while girls' preference orders are the following: 1: ACB, 2: CBA, 3: BAC.

1. (10 points) List all possible stable matching solutions in the given case. That is, each solution consists of a set of paris S. S is a stable matching.
2. (10 points) Point out the stable matching that can be derived from the boy-initiated (i.e., boys propose) G-S algorithm. Give a brief explanation.

Ans. (1) (A2, B3, C1) and (A1, B3, C2). (2) G-S algorithm can only derive (A2, B3, C1).

(**Problem 3**. 20 points) Consider the Interval Scheduling problem (page 116) again. Suppose we still have n requests with s_i and f_i for ith request's starting and finishing time, respectively. Now we have two persons A and B to select these requests. Each request can be assigned to one person only. The objective is the find a schedule that maximizes the sum of the number of requests assigned to A and B.

1. Suppose we apply the greedy algorithm first to find out assignment to A and then apply the algorithm again on the remaining requests for B. Apply this algorithm to the example shown on page 5 of Chapter 4 classnotes

   `https://cis.temple.edu/~jiewu/teaching/Spring2021/Chap4.pdf`

2. Will this approach achieve the maximization? If yes, please provide a proof; otherwise, give a brief explanation, followed by a counter example.

Ans. (1) $\{b, e, h\}$, $\{c, f\}$ (2) No. Counter example, a: [1, 3], b: [2, 4], c: [4, 5], and d: [3, 6]. Greedy will generate $\{a, c\}$, $\{b\}$. The optimal one is $\{a, d\}$, $\{b, c\}$.

(**Optimal greedy**: Apply the earliest finishing-time first for job selection. However, the assignment decision is made at time s_i, *scheduling point*, for the selected job i. If none is available at s_i, i is discarded; otherwise, i is assigned to the *best-fit* person with a larger finishing time of his/her last assigned job.)

(**Problem 4**. 20 points) Reconsider exercise 1 of Chapter 5 (page 246). You'd like to determine the $3n$ th smallest values among the set of $4n$ values. These values are equally partitioned and assigned to four sorted databases with n values in each.

1. Assuming $n = 2^k$, give an algorithm that finds the $3n$ th smallest values using at most $O(\log n)$ queries.

2. How would you revise your algorithm if $n \neq 2^k$. Briefly show how the algorithm handles when $n = 53$.

Ans. (1) Find the nth largest value (i.e., DBs are sorted in the decreasing order). The initial sample point is at the $\frac{n}{8}$th value for each of the four DBs. Remove one segment of $\frac{n}{8}$ with the largest sample. Overall, there are $4 \times \log n$ sample points applied in sequence: $\frac{n}{8}$, $\frac{n}{8}$, $\frac{n}{8}$, $\frac{n}{8}$, $\frac{n}{16}$, $\frac{n}{16}$, $\frac{n}{16}$, $\frac{n}{16}$, $\frac{n}{16}$, $\frac{n}{32}$, $\frac{n}{32}$, $\frac{n}{32}$, $\frac{n}{32}$, ... At each of the above sample point, there are at least $3n$ values that are below the four samples of four DBs. The overall complexity is $O(\log n)$. (2) Instead of using floor and ceiling functions, we use the binary code, e.g., 53=32+16+4+1 = 110101.

(**Problem 5**. 20 points) Suppose we make the following two changes to the RNA secondary structure (page 274):

(i) (No sharp turn) is changed to: if $(i, j) \in S$, then $i < j - c'$, where c' is a constant.

(v) (Limited nested pairs). There are no c pairs (c is a constant): (i_k, j_k) for $k = 1, 2, ..., c$ such that $i_1 < i_2 < ... < i_c < j_c < ... < j_2 < j_1$. Nested pairs less than c are allowed. Fig. 6.14 (page 276) in the textbook shows an example of $c = 5$ nested pairs.

1. Describe in plain English how you would change the dynamic programming structure.
2. Rewrite (6.13) (page 276) and rewrite the pseudo code on page 277.

Ans: Change 4 to c' and add one dimension c in the DP for a 3-D array build-up process.

$$OPT(i, j, k) = \max\{OPT(i, j-1, k), \max_t\{1 + OPT(i, t-1, k) + OPT(t+1, j-1, k-1)\}\}$$

where $k = 0, 1, ..., c - 1$.

(**Bonus problem**, 10 points) Briefly discuss why all boys are truthful in the boy-initiated G-S algorithm, while it is not the case for girls. That is, in some cases, girls do not follow the G-S matching rule for girls, but end up better matching. Use the given example in Problem 2 to illustrate.

Ans. (1). G-S is already male-optimal. (2).When A first proposes to 2, girl 2 **lies** by *rejecting the first suitor* (but runs a risk). A then goes after 1. B has no choice but goes after 3. Eventually, C will propose to 2 (ideal for 2). (3). Girl can also lie by *changing the preference order*: C proposes to 1 to have C1. Then B proposes to 1, but girl 1 **lies** and accepts B to form B1. The rejected C then proposes to his second 2 to form C2. Since 2 is no longer available, A proposes to 1, 1 accepts A to form A1. Finally, the rejected B proposes to 3 to form B3.