Homework #2 (due 1pm November 5, 2018 in BL 406)

(A quiz covering Chapters 6–9, 12, and 13 will be given in-class on November 1, 2018.)

1. Work on (a) Exercise 6.4-1 (page 160), (b) Problem 7.1(a): Hoare partition (page 185) and Quicksort discussed in class, and (c) Exercise 8.2-1 (page 196) based on the string (array of 16 characters): "*NTUEECSALGORITHM*". Please mark the two T's as T_1 and T_2, and the two E's as E_1 and E_2 according to their order in the input, and **show their positions during the processing**. For (c), assume you have only the 26 characters, $A, B, \ldots, Z$ and thus you may work on the array of the 26 characters.
2. Problem 6-3 (pages 167–168).
3. Exercise 8.2-4 (page 197).
4. (a) Exercise 8.3-1 (page 199); (b) Exercise 8.4-1 (page 204).
5. Problem 8-4 (pages 206–207).
6. Exercise 9.1-1 (page 215).
7. Exercise 9.3-1 (page 223).
8. Exercise 9.3-8 (page 223.).
9. Exercise 9.3-9 (pages 223–224).
10. Exercise 12.2-1 (a), (c), and (e) (page 293).
11. Problem 12-2 (page 304).
12. Search trees.
    (a) Give the binary search tree that results from successively inserting the keys 7, 2, 1, 5, 4, 6, 8, 9 into an initially empty tree.
    (b) Label each node in the tree with R or B denoting the respective colors RED and BLACK so that the tree is a legal red-block tree.
    (c) Give the red-black tree that results from inserting the key 3 into the tree of (b).
    (d) Give the red-black tree that results from deleting the key 9 from the tree of (c).
13. Problem 13-3 (page 333).
14. (a) Exercise 15.2-1 (page 378). (b) Exercise 15.4-1 (page 396). (c) Exercise 15.5-2 (page 404).
15. Exercise 15.4-6 (page 397).
16. Let $X = x_1x_2 \ldots x_m$ and $Y = y_1y_2 \ldots y_n$ be two character strings. This problem asks you to find the maximum common **substring** length for X and Y. Notice that substrings are required to be contiguous in the original strings. For example, *photograph* and *tomography* have common substrings *ph*, *to*, *ograph*, etc. The maximum common substring length is 6.
    (a) Figure 2 gives the computation of the maximum common substring length on the two strings, $ABAB$ and BAB, similar to the table used for computing the length of LCS in class. Only partial results are given. Please complete all the entries in the table.

(b) Dynamic programming can be used to find the longest common substring efficiently. The idea is to find length of the longest common **suffix** for all substrings of both strings. Suppose $\gamma = \alpha\beta$ is the concatenation of two strings; we say that β is a *suffix*. Find the optimal substructure for the longest common suffix problem (a recurrence relation for $LCSuff(X, Y, m, n)$).

(c) Define the longest common substring length $LCSubStr(X, Y, m, n)$ in term of $LCSuff(X, Y, i, j)$.

(d) Give a dynamic programming algorithm for solving this problem. What are the time and space complexity of your algorithm?

		A	B	A	B
	0	0	0	0	0
B	0	0			
A	0				
B	0				

Figure 1: Table for computing the maximum common substring length.

17. You are given a sequence of n circuit cells $C = < c_1, c_2, \ldots, c_n >$ in a single row with the **fixed** cell order from left to right, $c_1 c_2 \ldots c_n$, each cell c_i with its **bottom-left** coordinate x_i and the width w_i, and the minimum spacing ϕ_{c_i,c_j} for a pair of cells c_i and c_j, $i \neq j$. For example, Figure 2(a) shows an initial placement of four circuit cells.

You are ask to minimize the total length of placing the n cells, flipping or not flipping each cell. That is, cell c_i can have two orientations, **not flipped (unflipped)** and **flipped**, denoted by c_i^p and c_i^r, respectively. A cell flipping graph can be constructed to visualize the cell flipping problem, as shown in Figure 2(b) where the nodes f_i^p and f_i^r respectively represent the two orientations c_i^p and c_i^r of the cell c_i, and the number beside each edge between two nodes (i.e., edge weight) denotes the minimum spacing of the two corresponding cell boundaries.

(a) For the example shown in Figure 2, the initial unflipped cell placement gives the total row length of 40, with $w_1 = 6, w_2 = 4, w_3 = 8$, and $w_4 = 10$, and the minimum spacing $\phi_{c_1^p,c_2^p} = 3$, $\phi_{c_1^p,c_2^r} = 1$, and so on. Find the optimal cell flipping with the minimum total row length for these four cells c_1, c_2, c_3, and c_4. What is the optimal row length? Which cell(s) should be flipped to achieve the optimal solution?

(b) This problem exhibits the optimal substructure with overlapping subproblems. Explain the properties of (1) the optimal substructure, and (2) overlapping subproblems.

(c) Let f^* denote the optimal cell flipping, and T denote the cost function of the nodes f_i^p, f_i^r, and f^*; i.e., $T(f_i^p)$ gives the x-coordinate of the unflipped cell i. Find the recurrences for $T(f_i^\alpha)$, $\alpha \in \{p, r\}$, and $T(f^*)$. **(Hint: in terms of x_i, w_i, $T(f_i^\alpha)$, $\phi_{c_i^\alpha,c_j^\alpha}$, etc., with appropriate indices.)**

(d) Give a dynamic programming algorithm for solving this problem. What are the time and space complexity of your algorithm?

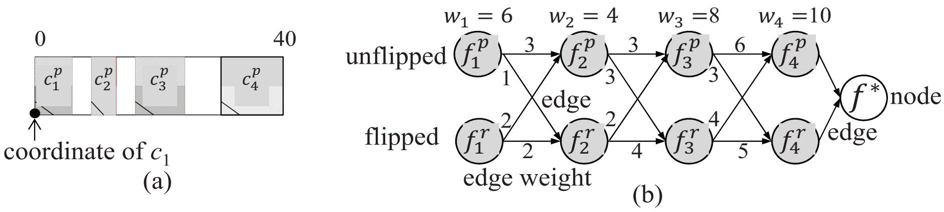

Figure 2: (a) An initial placement without any cell being flipped. (b) Cell flipping graph.

18. (DIY Problem) For this problem, you are asked to design a problem *set* related to Chapter(s) 6–9, 12, 13, and/or 15 and give a sample solution to your problem set. Grading on this problem will be based upon the *quality* of the designed problem as well as the *correctness* of your sample solution.