NATIONAL TAIWAN NORMAL UNIVERSITY
Department of Computer Science and Information Engineering

Introduction to Algorithms

Midterm Examination
Tuesday 04/12/2022

Instructions:

- This exam contains 6 problems, some with multiple parts. You have 110 minutes (10:20-12:10).

- This exam is closed book. No calculators or hand-held devices are permitted.

- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it.

- You may find the total score exceeds 100. Those extra points are bonus! Consider the time you need to solve for each problem, and make a good decision that will give you the maximal points in 110 minutes. The maximal score you can get is 100.

Good luck!

Problem #1 (20). Solve the following recurrences or the complexity of a function by giving tight Θ-notation bounds.

(a) $T(n) = 2T(n/2) + 1$

(b) $T(n) = T(n/3) + T(2n/3) + 3n$

(c) $T(n) = T(n-2) + T(n-1) + 1$

(d)

```
void j(int n) {
   if ( n == 1 ) {
      return;
   }
   j(n/2);
   j(n/2);
   int total = 0;
   for ( int i=0; i<n; i++ ) {
      total = total + i;
   }
}
```

Problem #2 (10). Divide-and-conquer algorithms usually perform more efficiently than brute-force algorithms. Please briefly describe what computations are eliminated in the following divide-and-conquer approaches if we compare them to the brute-force methods.

(a) Closest pair

(b) Merge sort

Problem #3 (10). True or false?

(a) $o(g(n)) \cap \omega(g(n))$ is an empty set.

(b) The worst case time complexity of Quick Sort is O(nlogn).

(c) An optimal binary search tree is more balanced than other binary search trees.

(d) The root of an optimal binary search tree always contains the key with the highest search probability.

(e) In the selection problem, the *Randomized-Select* algorithm is used to select an element randomly from an input array as the pivot to divide the array into sub-arrays. The average-case running time of the *Randomized-Select* algorithm is O(n).

Problem #4 (15).

(a) What is *stable sort*? **(5)**

(b) Please briefly describe why *radix sort* is required to perform digit-sort starting from least significant digit to most significant digit. **(5)**

(c) Continuing (b), please briefly describe why the digit-sort must be stable in order for radix sort to work correctly. **(5)**

Problem #5 (25). Let A be an array of n distinct integers. A pair $(A[i], A[j])$ is said to be an **inversion** if $i < j$ and $A[i] > A[j]$.

(a) Given $A = [8, 4, 1, 2]$, what is the number of inversions in A? **(5)**

(b) We can use Bubble Sort to find the number of inversions in a sequence. Below is the pseudo code of Bubble Sort. **(10)**

```
1 BubbleSort(A)
2     for i = 1 to A.length - 1
3         for j = A.length downto i + 1
4             if A[j] < A[j - 1]
5                 exchange A[j] with A[j-1]
```

Please explain why the number of bubble-sort swaps equals to the number of inversions.

(c) We can also modify the Merge Sort procedure to find the number of inversions in a sequence. It is an $O(n\log n)$ algorithm. Please describe this algorithm. **(10)**

Problem #6 (25). Given the following operations, the *edit distance* of two sequences X and Y is the minimal number of operations that transform X to Y.

Replace: set x_i to some other character
Insert: add a character into the sequence
Delete: remove a character from the sequence

(a) Please describe a brute-force approach for solving the problem. What is the time complexity? **(5)**

(b) What is the edit distance between $X = [cats]$ and $Y = [fast]$? **(5)**

(c) We can solve this problem by dynamic programming. Please write down the recurrence. **(5)**

(d) Continuing (c), please modify the recurrence to find the edit distance of two sequences if we allow only **Insert** and **Delete**. **(5)**

(e) Consider the following four variables:

$|X|$: the length of the first input sequence X
$|Y|$: the length of the second input sequence Y
dist: the edit distance between X and Y if we allow only **Insert** and **Delete**
$|LCS|$: the length of longest common sequence (LCS) of X and Y

Please write down an equation that relates these variables. **(5)**