CSE 100: Algorithm Design and Analysis
Midterm 1

Fall 2022

- Exam time: noon-1:15pm
- This exam allows one page of hand-written cheat sheets. One note paper (front and back, P7 & P8) is prepared. Please submit your note paper with your exam sheet after the exam.
- There are 7 problems of 100 points in total. You can earn some partial points if you show progress even if you can't solve problems completely.
- Please make your answers concise as well as precise.
- Please fill in the following blanks for your information. (For the following pages, there are also spaces to write your name in case you tear that page down accidentally.)
- Name:
- Catcard Number:
- Lab Num:(E.g. 02, 03, 04, 05, 06, 07)

1. (18 points) For each of the claims, decide if it is true or false. *No* explanation is needed.

   (a) (2 points) $n \log n = O(n^2)$.
   Sol. True

   (b) (2 points) $\log \log n = O(\log n)$.
   Sol. True

   (c) (2 points) if $T(n) = T(n/2) + O(n)$, $T(n) = O(n)$.
   Sol. True

   (d) (2 points) if $T(n) = 2T(n/2) + O(n)$, $T(n) = O(n \log n)$.
   Sol. True

   (e) (2 points) if $T(n) = 4T(n/2) + O(n)$, $T(n) = O(n^2)$.
   Sol. True

   (f) (2 points) $100^{100} = \Theta(1)$.
   Sol. True

   (g) (2 points) if $f = O(g)$, then $g = \Omega(f)$.
   Sol. True

   (h) (2 points) if an algorithm's worst case running time is $O(n^2)$, it means that the running time is $\Theta(n^2)$ for all inputs of size n.
   Sol. False

   (i) (2 points) $100 + 200n + 300n^2 = \Theta(n^2)$
   Sol. True

   Sol. Many of them straight from Discussion 2 & 3

2. (9 points) The following is a pseudocode of Insertion-sort. For instance $A[1\dots 8] = \langle 7,4,2,9,4,3,1,6\rangle$, what is $A[1\dots 8]$ just before the for-loop starts for $j = 5$?

```
Insertion-Sort(A)
for j = 2 to A.length
   key = A[j]
   // Insert A[j] into the sorted sequence A[1...j - 1].
   i = j - 1
   while i > 0 and A[i] > key
     A[i + 1] = A[i]
     i = i - 1
   A[i + 1] = key
```

Sol. $\langle 2,4,7,9,4,3,1,6\rangle$. If the answer is correct for $j = 4$ or $j = 6$, take off 2 pts.

3. (12 points) The following is a pseudocode of the naive divide-and-conquer algorithm for matrix multiplication. Here, partitioning a n by n matrix means partitioning it into four $n/2$ by $n/2$ (sub-)matrices.

```
SQUARE-MATRIX-MULTIPLY-RECURSIVE(A, B)
n = A.rows
let C be a new n × n matrix
if n == 1
    c11 = a11 · b11
else partition A, B, and C as in equations (4.9)
    C11 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B11)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B21)
    C12 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B12)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B22)
    C21 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B11)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B21)
    C22 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B12)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B22)
return C
```

Give the recurrence for the running time of Square-Matrix-Multiply-Recursive and solve it. We let $T(n)$ denote the running time when input matrices are n by n. No need to show how you solved it. Just state the final result along with the recurrence.

Sol. $T(n) = 8T(n/2) + \Theta(n^2)$. (4 pts)
It is okay to use O in place of Θ.

Sol. $T(n) = \Theta(n^3)$. (4 pts)
It is okay to use O in place of Θ.

This is straight from the Lecture Slides (L06-Matrix Multiplication), Lecture Notes (L06-Notes) and/or CLRS pg. 76-78

4. (12 points) Let $T(n)$ denote the running time of Merge-sort on input of size n. In the following you can omit floor or ceiling.

```
MERGE-SORT(A, p, r)
1  if p < r
2      q = ⌊(p + r)/2⌋
3      MERGE-SORT(A, p, q)
4      MERGE-SORT(A, q + 1, r)
5      MERGE(A, p, q, r)
```

What is the running time of Line 3 (of Merge-Sort)?

What is the running time of Line 4 (of Merge-Sort)?

What is the running time of Line 5 (of Merge-Sort)?

Sol. $T(n/2)$, $T(n/2)$, $O(n)$ (or $\Theta(n)$). 3, 3, 6 points, respectively.

5. (9 points) Formally prove that $50n + 15 = O(n^2)$ using the definition of $O(\cdot)$.

Sol.
One possible answer: $50n + 15 \leq 65n^2$ for all $n \geq 1$.

BOTH Questions are straight from the Lecture Slides (L03 and L04), Lecture Notes (L03 and L04-Notes) and/or CLRS

6. (16 points) Give a *pseudocode* of Heap-sort. For simplicity, you can assume that the heap A you're given is *already* a max-heap. You can use the function Max-Heapify(i) as a subprocedure. Recall that the function Max-Heapify(i) makes the subtree rooted at node i a max-heap, *if both* the left and right subtrees of node i are max-heaps. You can use $A.heapsize$ to denote the current heap size. If you can't give a pseudocode, you can describe the Heap-sort algorithm in words, but you will lose some points.

Sol. Pseudocode for HeapSort:

```
HeapSort(A)
BUILD-MAX-HEAP(A) [optional, since A is already a max-heap!]
for i = A.length downto to 2
   exchange A[1] with A[i]
   A.heapsize = A.heapsize - 1
   MAX-HEAPIFY(A,1)
```

See CLRS page 159-160 for more details.

This is straight from the Lecture Slides (L07-Heaps), Lecture Notes (L07-Notes, pg. 6-7) and/or CLRS (pg. 159-160).

7. (24 points) The k-SELECT problem is to find the k-th smallest number in an array. The pseudocodes of the SELECT function and the PARTITION function are given as follows. Given A=[8, 9, 2, 6, 7, 1, 5], please answer the following questions. Please note that A[0]=8.

- SELECT(A, p=k):
  - p = CHOOSEPIVOT(A)
  - L, A[p], R = PARTITION(A,p)
  - **If** len(L) = k - 1:
    - **Return** A[p]
  - **Else If** len(L) > k - 1:
    - **Return** SELECT(L, k)
  - **Else if** len(L) < k - 1:
    - return SELECT(R, k – len(L) – 1)

- PARTITION(A, p):
  - L = new array
  - R = new array
  - **For** i=1,…,n:
    - **If** i==p:
      - continue
    - **else If** A[i] <= A[p]:
      - L.append(A[i])
    - **Else if** A[i] > A[p]:
      - R.append(A[i])
  - **Return** L, A[p], R

(a) (4 points) What is the return result of SELECT(A, 1)?

(b) (4 points) What is the return result of SELECT(A, 3)?

(c) (4 points) What is the return result of SELECT(A, 7)?

(d) (6 points) For SELECT(A, 3), k=3. If the return result of CHOOSEPIVOT is 2, which one is true, $len(L) > k - 1$ or $len(L) < k - 1$?

(e) (6 points) For SELECT(A, 3), k=3. If the return result of CHOOSEPIVOT is 2, which one is true, $len(L) > k - 1$ or $len(L) < k - 1$?

Sol.

(a) 1

(b) 5

(c) 9

(d) $len(L) < k - 1$. Since the pivot is 2, A[2] =2. L=[1], and R=[5, 6, 7, 8, 9]. Thus Len(L)=1.

(e) A bonus of 6 points to all the students.