CS 6363: Design and Analysis of Algorithms
Exam #1 October 8, 2018
Professor D.T. Huynh

Student Name:	KEY

General Remarks. This exam comprises 4 problems:

Problem ♯1 is assigned 25 points,

Problem ♯2 is assigned 25 points,

Problem ♯3 is assigned 25 points, and

Problem ♯4 is assigned 25 points.

Thus, the maximum score is 100 points.

Unless explicitly stated, *no correctness proofs* are required for your algorithms and (time) complexity means worst-case complexity.

Provide clean answers on the exam booklet. Use aditional paper only when necessary.

This is a **closed-book** exam

Exam time: 10:00 – 11:20 am

Good Luck!

♯1	♯2	♯3	♯4	Total

Problem ♯ 1.

1. Compare the order of magnitude of the following pair of functions. In each case determine whether $f(n) = O(g(n)), f(n) = \Omega(g(n))$, and/or $f(n) = \Theta(g(n))$. Justify your answers!

   (a) $f(n) = 120n^{1.01} + nlgn, \quad g(n) = nlg^k n$ where $k > 1$ is a fixed integer.

   (b) $f(n) = (nlglgn)^{lgn}, \quad g(n) = (1.5)^n$

2. State the Master Theorem and use it to derive a tight bound for $T(n) = 3T(n/2)+n^2 lgn$.

1. (a) $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\cancel{n}(120 n^{.01} + lgn)}{\cancel{n}\, lg^k n} = \infty$$

$$\Rightarrow \boxed{f(n) = \Omega(g(n))}$$

10

(b) $$lg\, f(n) = lg(n)\, lg(n\, lg\, lg\, n) = lgn\,(lgn + lglglgn)$$
$$= lg^2 n + lgn.\, lglglgn$$
$$lg\, g(n) = n.\, lg\, 1.5$$
$$\Rightarrow \lim_{n\to\infty} \frac{lg\, f(n)}{lg\, g(n)} = \lim_{n\to\infty}\left(\frac{lg^2 n}{n.\,lg 1.5} + \frac{lgn\, lglglgn}{n.\,lg\,1.5}\right) = 0$$
$$\Rightarrow f(n) = o(g(n)) \text{ or } O(g(n))$$

(2) MT: see class notes 10

$$T(n) = 3\,T(\tfrac{n}{2}) + n^2 lgn \Rightarrow a=3,\ b=2,\ f(n) = n^2 lgn$$
$$\Rightarrow n^{lg_b a} = n^{lg_2 3} \Rightarrow f(n) = \Omega(n^{lg_b a + \varepsilon})$$

Moreover, $$af(\tfrac{n}{b}) = 3.(\tfrac{n}{2})^2 lg(\tfrac{n}{2})$$
$$= \tfrac{3}{4} n^2 (lgn - lg2) \leq \tfrac{3}{4} n^2 lgn$$

$$\Rightarrow \boxed{T(n) = \Theta(n^2 lgn)}$$ 5

Problem ♯ 2. (Bubblesort)

```
BUBBLESORT(A)
for $i = 1$ to $A.length - 1$                    $\Theta(n)$
    for $j = A.length$ downto $i + 1$            $\Theta(n-i)$
        if $A[j] < A[j-1]$
            swap $A[j]$ with $A[j-1]$
```

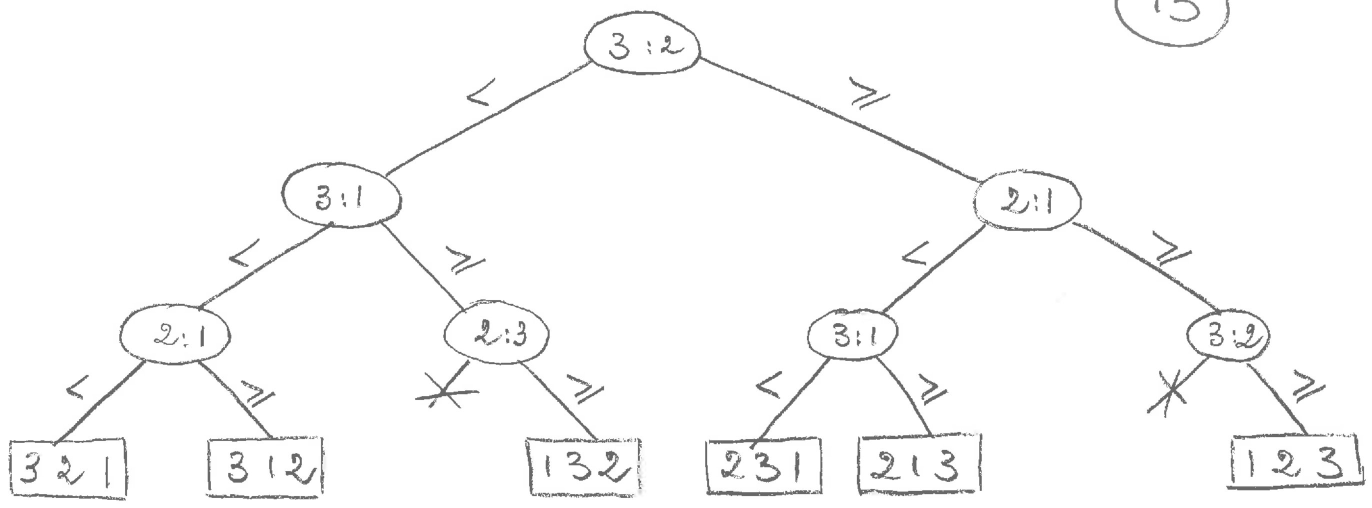
1. Analyze the worst-case complexity of BUBBLESORT using the Θ notation.

$$\sum_{i=1}^{n-1} \Theta(n-i) = \Theta(n^2)$$

12

2. Draw the decision tree obtained from BUBBLESORT for an array $A[1..3]$ of size 3.

13



Problem ♯ 3.

1. For the chained-matrix multiplication problem consider the matrices A, B, C, D with dimensions $20 \times 1, 1 \times 25, 25 \times 10$ and 10×15, respectively. Perform the dynamic-progarmming algorithm for the chained-matrix multiplication problem with A, B, C, D in the input. (Fill out the entries in the following table and display the details of the last iteration for computing m_{14}.)

8

$m_{14} = 700$			
$m_{13} = 450$	$m_{24} = 400$		
$m_{12} = 500$	$m_{23} = 250$	$m_{34} = 3750$	
$m_{11} = 0$	$m_{22} = 0$	$m_{33} = 0$	$m_{44} = 0$

Computing m_{14}

$$\begin{tikzcd} & m_{14} \arrow[dl, no head] \arrow[dr, no head, "k=1"] & \\ m_{11} & & m_{24} \end{tikzcd}$$

$0 + 400$
$+ 20 \times 1 \times 15$
$= 700$
↑ minimum

$$\begin{tikzcd} & m_{14} \arrow[dl, no head] \arrow[dr, no head, "k=2"] & \\ m_{12} & & m_{34} \end{tikzcd}$$

$500 + 3750$
$+ 20 \times 25 \times 15$
> 700

$$\begin{tikzcd} & m_{14} \arrow[dl, no head] \arrow[dr, no head, "k=3"] & \\ m_{13} & & m_{44} \end{tikzcd}$$

$450 + 0$
$+ 20 \times 10 \times 15$
> 700

5

2. To actually compute an optimal order what kind of information has to be recorded while executing the algorithm? Modify the dynamic-progamming algorithm so that such information can be recorded and analyze its running time.

In $m_{i,i+s} = \min_{i \le k < i+s} (m_{ik} + m_{k+1,j} + d_{i-1} \times d_k \times d_j)$

record k_0 s.t. it gives minimum.

```
MAT_MULT ( d [0..n] , n )
  Let m[1..n, 1..n] , k[1..n, 1..n] be int. arrays
  for i = 1 to n do  m[i,i] = 0 ;
  for s = 1 to n-1 do
      for i = 1 to n-s do
          j = i+s ;
          min = ∞ ;
          for k = i to j-1 do
              q = m[i,k] + m[k+1,j] + $d_{i-1} \times d_k \times d_j$
              if q < min then
                  min = q
                  k[i,j] = k   /* record k */
```

8

$O(n^3)$

4

Problem ♯ 4. Let $A[1..n]$ be an array of integers. An element $x \in A$ is said to be *significant* if it appears at least $\lceil n/4 \rceil$ times in A. The goal is to design an $O(n)$ algorithm to compute all significant elements in $A[1..n]$ if they exist, assuming the linear SELECTION algorithm as a black box.

1. The linear SELECTION algorithm assumes all elements in the input array are distinct. In order to apply SELECTION, show how you can modify the elements in the array A to obtain a new array B such that all elements of B are distinct. Your algorithm should be linear.

```
Let $k = \lceil \log n \rceil$
    for $i = 1$ to $n$ do
        $B[i] = A[i] * 2^k + i$
```

$\boxed{10}$

Clearly, the algor. runs in $O(n)$ steps

2. Noting that in case A is sorted a significant element will appear *consecutively* at least $\lceil n/4 \rceil$ times, show how to use SELECTION on B to compute all significant elements in A in linear time. Analyze the running time of your algorithm.

$\boxed{15}$

Perform SELECTION on $B[1..n]$ to find

- $\lceil \frac{n}{4} \rceil$ smallest element a in B
- $\lceil \frac{n}{2} \rceil$ ——— b in B
- $\lceil \frac{3n}{4} \rceil$ ——— c in B
- the max. element d in B.

Let a' (b', c', d') be $a \div 2^k$ $(b \div 2^k, c \div 2^k, d \div 2^k)$

Now count # of occurrences of a', b', c', d' in A and output if it's significant.

SELECTION and counting are both linear

$\Longrightarrow$ Algor. is $O(n)$.