LE/EECS 3101 A
Design & Analysis of Algorithms

Midterm

Shahin Kamali

York University

October 19th, 2023

Write your name and student id here:

"To deny people their human rights is to challenge their very humanity ..." *Nelson Mandela*

- **Do not open this booklet until instructed**.
- You are NOT allowed to use any printed/written material. A "cheat page" is provided at the end of the exam. It is OK to take the staples off, but make sure to **submit all sheets**, including the cheat sheet.
- Please **turn off your cell phones** and put them in your bags.
- Calculators are not needed but you can use simple calculators with no memory.
- Manage your time. We start the exam at 4:00 and end the exam at 5:30. **You have 90 minutes**. **Don't waste too much time on a single question. It is a long exam, and your time is limited.**
- To save trees, the exam is printed **double-sided**. There are **10 pages**, including this cover page, the cheat page, and two extra blank pages (use them if you need more space). You must submit ALL pages.
- If you find the exam too long/hard (which is likely), do not panic. The marks will be scaled so that the highest mark gets the full mark.

1. Short Answer Questions (24 marks)

Provide your short answers in the provided boxes. There is no need to justify your answers. **Notes:** all parts have 2 marks except the last two which have 3 marks.

1. True or False: Multiplying two matrices A and B, each of size $n \times n$, takes $\Omega(n^2)$. True **Answer:**

   You need to read the two matrices to multiply them, and it asymptotically takes n^2 time.

2. True or False: $\log(n^{2023}) \in \Theta(\log n^{1917})$. True **Answer:** We have $\log(n^{2023}) = 2023 \log n$ and $\log n^{1917} = 1917 \log n$, and the two functions are a constant factor away from each other.

3. True or false: In any recursion tree, the number of leaves is larger than or equal to the number of internal nodes. False **Answer:** For example, in the binary search recursion, there is only one recursive call and thus one leaf in the recursion tree.

4. True or False: $\frac{\sqrt{n}}{\log n} \in \omega(\log^{2023} n)$ True **Answer:** For any positive values of ϵ and k, as long as they are constants (independent of n), we have $n^\epsilon \in \omega(\log^k n)$. In particular, $n^{1/2} \in \omega(\log^{2024} n)$ which confirms the statement of this question is correct.

5. True or False: The following function has complexity $\Theta(n^2 \log n)$: False

$$f(n) = \begin{cases} 100\log(n), & \text{for } 0 \leq n \leq 1402 \\ n^2 \log n + \frac{1}{100}\log^{2023} n, & \text{for } 1402 \leq n \leq 2023 \\ 2n\log n + \log n^{2023}, & \text{for } n > 2023 \end{cases}$$

   Answer: For asymptotic analysis, we only care for arbitrary large values of n; here we have $f(n) = \Theta(n \log n)$. We cannot state that the complexity is max taken over the three given functions.

6. True or False: if $\log(f(n)) \in \Theta(\log(g(n)))$ then $f(n) \in \Theta(g(n))$. False **Answer:** Assume $f(n) = 2^{2n}$ and $g(n) = 2^n$. We have $\log f(n) \in \Theta(\log g(n))$. But $f(n) = 2^{2n} = 2^n \times 2^n$ which is asymptotically larger than $c \times 2^n$ for constant c. Here $f(n) \in \omega(g(n))$.

7. True or False: There are heaps of size n for which ExtractMax operation takes $\Theta(n)$ in the worst case. False

   Answer: False. ExtractMax takes $O(\log n)$ in the worst case.

8. True or False: Strassen's Algorithm for matrix multiplication makes 7 recursive calls, and its time complexity for $n > 2$ is given by $T(n) = 7T(n/2) + \Theta(n^2)$. True

9. True or False: If we select the pivot as the first element, the worst case running time of Quick-Sort and Quick-Select will be the same. True **Answer:** Both algorithm will run in $O(n^2)$ in the worst-case (with this naive pivot-selection).

10. Assume $T(1) = 2023$ and $T(n) = 25T(n/5) + n^2 \log n$. Give an expression for the run-time of $T(n)$ using Θ notation. $\Theta(n^2 \log^2 n)$ **Answer:** Case two of the Master theorem. We have $n^{\log_b a} = n^{\log_5 25} = 2$. So we have $T(n) = n^2 \log^2 n$.

11. Assume $T(1) = 1402$ and $T(n) = 4T(n/8) + 3n^3$. Give an expression for the number of leaves in the recursion tree of T using Θ notation. $\Theta(n^{\log_8 4})$ **Answer:** At each level, the number of nodes increases by a factor of 4; since we divide by 8, there are $\log_8 n$ levels. Therefore, there will be $4^{\log_8 n} = n^{\log_8 4}$ leaves.

2. Asymptotic Analysis (4 marks)

Provide a complete proof of the following statement from first principles (i.e., using the original definitions of order notation).

$$n^{1.5} \in \omega(n^{0.49+\cos n})$$

Answer: We need to show that for any M, there is some n_0 such that for $n > n_0$ it holds that $n^{1.5} > Mn^{0.49+\cos n}$. Given that $\cos n < 1$, we can write $n^{0.49+\cos n} < n^{1.49}$. Therefore, it suffices to prove $n^{1.5} > Mn^{1.49}$, which is equivalent to $n^{0.01} < M$ or $n > M^{100}$. That is, it suffices to have $n_0 \geq M^{100}$.

3. Loop Analysis (5 marks)

Consider the following pseudocode:

```
foo(n)
1.  for i = 1 to n do
2.      k ← 1
3.      for j = 1 to n do
4.          k ← i * k
5.      while k > 1 do
6.          k ← k/2
7.  return i
```

What is the worst-case running time of *foo*(n)?
Express your answer using Θ-notation in terms of n, and be as precise as possible.
Show your work in the space below.

Answer: Lines 2-4 clearly take $\Theta(n)$. At the end of Line 3, the value of k is i^n. Therefore, Lines 4-5 run in $c \log i^n = cn \log i$ for constant c (which dominates the time complexity of Lines 2-4). That is, iteration i of the for loop at line 1 takes $c'n \log i$. Summing over all iterations, the total time complexity will be $c'n \sum \log i = \Theta(n^2 \log n)$.

4. Divide & Conquer (10 marks)

Given an array A of integers with size n, devise a **divide and conquer** algorithm that runs in $\Theta(n \log n)$ and reports the length of the longest contiguous subarray of A formed by the same number. For example, for $A = [1, 0, 0, 3, 4, 4, 2, 2, 0, 0, 0, 3]$, the output should be 3, because there is a subarray formed by 3 consecutive 0's.

You need to present the body of a procedure with header LongestSameSubArray(A, lo, hi) (you are expected to know the meaning of lo and hi). The first call is LongestSameSubArray$(A, 0, n-1)$. There is no need to analyze the time complexity.

Answer:

```
LongestSameSubArray(A, lo, hi)
    if (lo = hi)
        return 1
    mid ← (lo + hi)/2
    option1 ← LongestSameSubArray(A, lo, mid)
    option2 ← LongestSameSubArray(A, mid + 1, hi)
    option3 ← helper(A, lo, mid, hi)
    result ← max{option1, option2, option3}
    return result
```

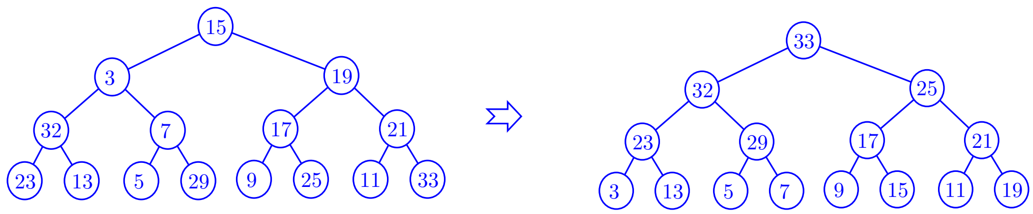
```
helper(A, lo, mid, hi)
    result ← 1.
    j ← mid − 1
    while (j ≥ low) and (A[j] = A[j + 1])
        result ← result + 1
        j ← j − 1
    j ← mid
    while (j ≤ hi − 1) and (A[j] = A[j + 1])
        result ← result + 1
        j ← j + 1
    return result
```

5. Heap Operations (6 marks)

Consider array $A = [15, 3, 19, 32, 7, 17, 21, 23, 13, 5, 29, 9, 25, 11, 33]$.

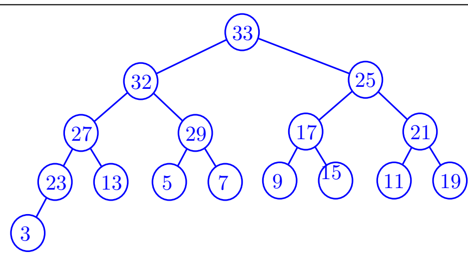
a) Apply the Heapify procedure on A to form a Max heap. Show the resulting tree in the space below.

Answer:



b) On the heap formed after the Heapify operation, apply operation *insert(27)*. Show the resulting tree in the space below.

Answer:



6. Median-of-Five Variant (8 marks)

The **trian** of k numbers is the $(k/3)$'th smallest number. For example, the trian of $\{7, 3, 8, 4, 1, 5\}$, where $k = 6$, is 3.

Consider the following variant of Median-of-Five algorithm. As before, blocks have size 5 and we find their medians. However, instead of selecting the pivot as the median of the $n/5$ medians, we select it as trian of the $n/5$ medians.

a) Follow the same steps as in the slides to derive a recursive formula for the time complexity $T(n)$ of this algorithm.

Answer: As before, finding the median of blocks and partitioning takes $\Theta(n)$. Finding trian is an instance of the selection problem, which takes $T(n/5)$. It remains to find the maximum size of the recursion of the qiuckSelect. Note that one third of blocks, that is $1/3 \times n/5 = n/15$ have their median and two other elements smaller than the pivot. That is, at least $3n/15 = n/5$ numbers are smaller than the pivot. So, when we recurs on right, the size of the recursion is at most $4n/5$. On the other hand, two-third of blocks, that is $2/3 \times n/5 = 2n/15$ have their median and two elements larger than the pivot. That is, at least $6n/15$ elements are larger than the pivot. So, when we recurs on the left, the size of the recursion is at most $9n/15$. Note that, in the worst case, the recursive call is on the right and of size $4n/5$. In this case, for $n > 1$, we can write

$$T(n) \leq \underbrace{T(n/5)}_{\text{finding trian of meidans}} + \underbrace{cn}_{selection} + \underbrace{T(4n/5)}_{\text{size of recursion}}$$

.

b) Try to solve the recursion by guessing that $T(n) \in O(n)$. Follow the same steps as in the slides and indicate whether we can state $T(n) \in O(n)$.

Answer: Let's guess $T(n) \in O(n)$ and use strong induction to prove it. We should prove there is a value M s.t. $T(n) \leq Mn$ for all $n \geq 1$. For the base we have $T(1) = d \leq M$ as long as $M \geq d$. For any value of n we can state:

$T(n) \leq T(n/5) + T(4n/5) + cn$ (from above recursion)

$\leq M \cdot n/5 + M \cdot 4n/5 + cn$ (induction hypothesis)

$= (M + c)n$

Note that we cannot show that $(M + c) \leq M$ for any value of M. So, following the same steps does Not give us the same result, i.e., we could Not prove that $T(n) \in O(n)$.

The following space is intentionally left blank. Use it if you need more space for your answers or draft work. Remember to submit this page.

You can consult the following "cheat page" in answering your questions. Not all material presented here is necessary to answer the exam questions. Remember to submit this page.

$$x^a x^b = x^{a+b}$$
$$(x^a)^b = x^{ab}$$
$$x^n + x^n = 2x^n \neq x^{2n}$$
$$x^n * x^n = x^{2n}$$
$$x^0 = 1$$
$$x^{-a} = \frac{1}{x^a}$$
$$\frac{x^a}{x^b} = x^{a-b}$$
$$(xy)^a = x^a y^a$$
$$2^n + 2^n = 2^{n+1}$$

$$a^{\log_b n} = n^{\log_b a}$$

$$\log_b(M) = X \leftrightarrow b^X = M$$
$$\log_b(b) = 1$$
$$\log_b(1) = 0$$
$$\log_b(0) \to \text{not defined}$$
$$\log_b(b^k) = k$$
$$b^{\log_b(M)} = M$$
$$\text{if } \log_b(M) = \log_b(N), \text{ then } M = N$$
$$\log_b(M^k) = k \cdot \log_b(M)$$
$$b^{\log_b(M)} = M$$
$$\log_b\left(\frac{1}{M}\right) = \log_b(M^{-1}) = -1 \cdot \log_b(M)$$
$$\log_b\left(\frac{M}{N}\right) = \log_b(M) - \log_b(N)$$
$$\log_b(M \cdot N) = \log_b(M) + \log_b(N)$$
$$\log_b(M) = \frac{\log_c(M)}{\log_c(b)}$$

$$\sum_{i=1}^{n} 1 = n$$
$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$$
$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3$$
$$\sum_{i=1}^{n} i^k = 1^k + 2^k + \cdots + n^k = \frac{1}{k+1}n^{k+1}$$
$$\sum_{i=0}^{n} a^i = 1 + a + \cdots + a^n = \frac{a^{n+1}-1}{a-1}(a \neq 1); \quad \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$
$$\sum_{i=1}^{n} i2^i = 1 \cdot 2 + 2 \cdot 2^2 + \cdots + n \cdot 2^n = (n-1)2^{n+1} + 2$$
$$\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma, \text{ where } \gamma \approx 0.5772\ldots \text{ (Euler's Constant)}$$
$$\sum_{i=1}^{n} \lg i \approx n \lg n$$

$$T(n) = \begin{cases} a\,T\left(\frac{n}{b}\right) + f(n) & \text{if } n > 1 \\ d & \text{if } n = 1. \end{cases}$$

$(a \geq 1,\ b > 1, \text{ and } f(n) > 0)$

- Compare $f(n)$ and $n^{\log_b a}$
- Case 1: if $f(n) \in O(n^{\log_b a - \epsilon})$, then $T(n) \in \Theta(n^{\log_b a})$
- Case 2: if $f(n) \in \Theta(n^{\log_b a}(\log n)^k)$ for some non-negative k then $T(n) \in \Theta(f(n)\log n) = \Theta(n^{\log_b a}(\log n)^{k+1})$
- Case 3: if $f(n) \in \Omega(n^{\log_b a + \epsilon})$ and if $af(n/b) \leq cf(n)$ for **some constant** $c < 1$ (regularity condition), then $T(n) \in \Theta(f(n))$

The following space is intentionally left blank. Use it if you need more space for your answers or draft work. Remember to submit this page.