CSCI 3104 Quiz 3

Jonathan Phouminh

TOTAL POINTS

14 / 15

QUESTION 1

1 1 / 2

- **- 0 pts** Correct
- ✓ **- 1 pts Missing correct answer(s)**
- **- 1 pts** Circled incorrect answer(s)
- **+ 0.5 pts** Circled a correct answer, but still had incorrect answers circle or correct answers omitted.
- **- 0.5 pts** Read the question carefully. Circle the comparisons that will NOT happen.
- **- 2 pts** Circled all four answers. Review binary search.

A[5] will never be evaluated. Note that (5+8)/2 == 6 (as we round down), so A[6] is evaluated next.

QUESTION 2

2 5 / 5

- ✓ **+ 3 pts Pivots selected properly to alternate between creating 2 n/2 splits and a single n-1 split**
- ✓ **+ 0.5 pts Swaps chosen pivot to end before starting partition**
- ✓ **+ 0.5 pts No unneeded swaps at each step (e.g. swapping two numbers that are both smaller than the pivot)**
- ✓ **+ 1 pts Correct sizes of branches in the tree**
- **+ 0 pts** Click here to replace this description.

QUESTION 3

3 3 / 3

- ✓ **+ 3 pts Correct with right explanation.**
- **+ 1.5 pts** Incorrect answer but with correct first recursive call.
- **+ 0 pts** Wrong answer with wrong process.
- **+ 0 pts** No answer
- **+ 2 pts** Incorrect(or correct) answer with minor errors.
- **+ 1 pts** Right answer with wrong explanation or wrong answer with partial right steps.

QUESTION 4

4 3 / 3

- ✓ **+ 3 pts Correct**
- **+ 2.5 pts** Please point out that it is the worst case.
- **+ 1.5 pts** It will only be the worst case.
- **+ 1.5 pts** It should be the worst case.
- **+ 1.5 pts** Explanation is not good/correct enough.
- **+ 0 pts** No answer.
- **+ 0 pts** Not correct.

QUESTION 5

5 2 / 2

- ✓ **+ 1 pts Correct recurrence**
- ✓ **+ 1 pts Correct use of the Master Theorem or Other Method**
- **+ 0 pts** Incorrect, insufficient work, or did not answer the question
- **+ 0.5 pts** Spell out more clearly how you are using the Master Theorem.
- **+ 0.5 pts** Partially correct recurrence
- **+ 0.5 pts** Partially correct use of the Master Theorem or other method.

Note that you should have O(n^3) and not n^3 in the recurrence. You don't know that finding the median takes n^3 steps, rather it takes O(n^3) steps. Also, O(n^3) absorbs the linear term.

Name: Jonathan Phaminh

ID: 106054641

CSCI 3104, Algorithms
Quiz 3' : 15 points total

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

Instructions: This quiz is open book and open note, but an individual effort. Electronic devices are not allowed on your person (including in your pocket). Possession of such electronics is grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show all work to receive full credit.**

Please provide these:
Left neighbor name :
Right neighbor name :

We provide the Master Theorem for your reference.

Master Theorem: Suppose $T(n) = aT(n/b) + f(n)$, where $a \geq 1$ and $b > 1$.

(a) If there exists $c < \log_b(a)$ such that $f(n) \in \Theta(n^c)$, then $T(n) \in \Theta(n^{\log_b(a)})$.

(b) If $f(n) \in \Theta(n^{\log_b(a)})$, then $T(n) \in \Theta(n^{\log_b(a)} \log(n))$.

(c) If $f(n) \in \Theta(n^c)$, where $c > \log_b(a)$, then $T(n) \in \Theta(f(n))$.

Name:

ID:

CSCI 3104, Algorithms
Quiz 3' : 15 points total

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

Answer the following questions. Justify each of your answers with a **short** *explanation. You won't receive any credit on the solutions that lack any explanation. You don't have to formally prove it.*

1. (2 pts) Consider the following binary search algorithm, where integers are rounded as in C++.

```
binarySearch(A[0, ..., n-1], int key, int left, int right){
    if right < left:
        return -1

    mid := (left + right)/2
    if A[mid] == key:
        return mid
    else if A[mid] < key:
        return binarySerach(A, key, mid+1, right)
    return binarySearch(A, key, left, mid-1)
}
```

$\frac{7+8}{2}$

$\frac{0+9}{2} \approx 4$

mid+1 = 5

$\frac{5+8}{2} = \frac{13}{2}$

0 4 5 mid mid 8

Let `A = [1, 3, 5, 7, 9, 11, 13, 15, 17]`. Suppose we invoke:

`binarySearch(A, 15, 0, len(A)-1).`

key left right

Which of the following comparisons would **NOT** happen? Circle all that apply.

(a) $A[5] == 15$

(b) $A[1] == 15$

(c) $A[7] == 15$

(d) $A[6] == 15$

$\frac{0+8}{2} = 4$

$\frac{5+8}{2} = \frac{13}{2} \approx 6$

Name:

ID:

CSCI 3104, Algorithms
Quiz 3' : 15 points total

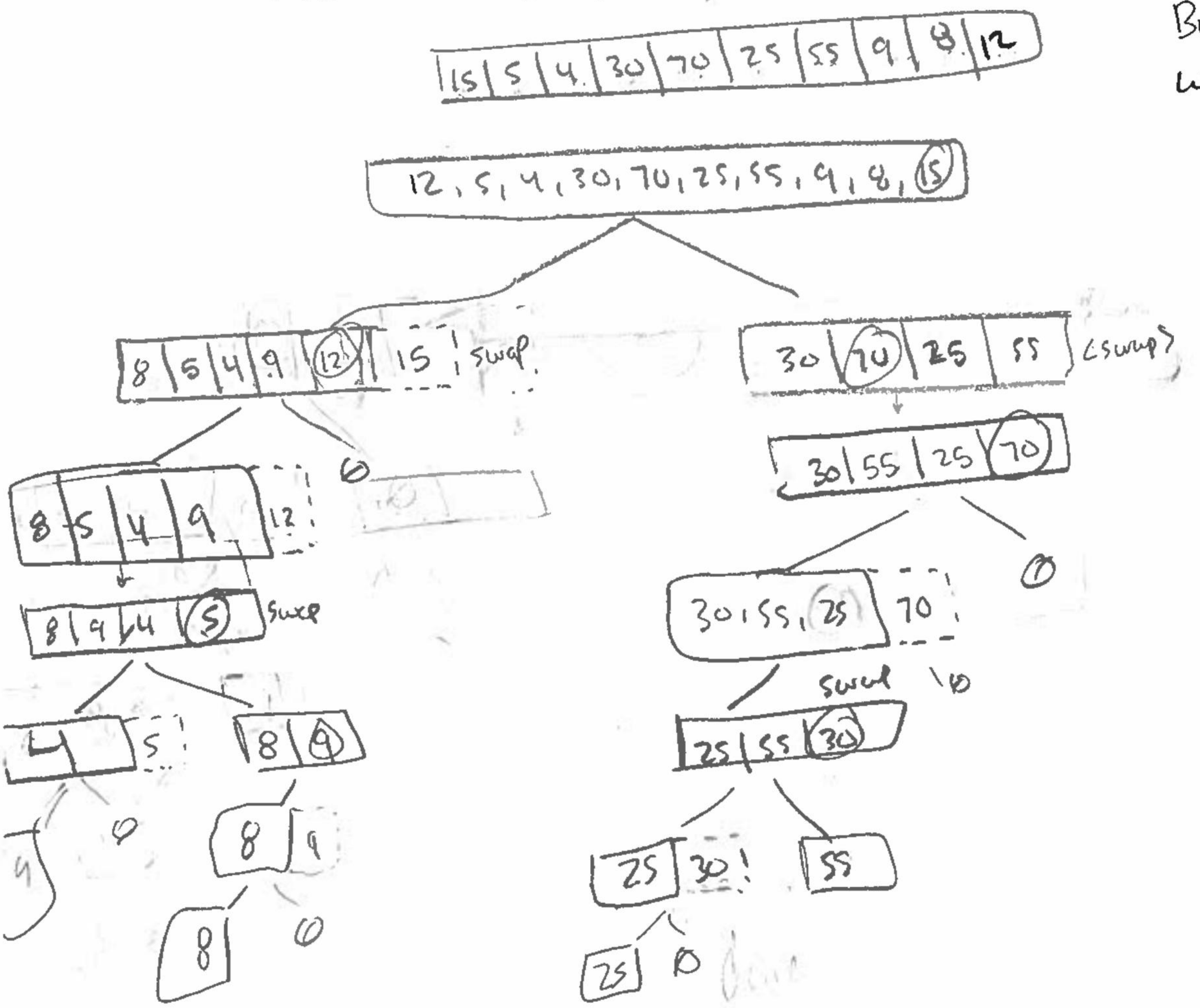
Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

4, 5, 8, 9, 12, 15, 25, 30, 55, 70

2. (5 pts) Assume we have a version of QuickSort with an alternating partitioning scheme which selects the best and worst pivot in an alternating manner. Draw the tree of recursive calls for given input array A=[15, 5, 4, 30, 70, 25, 55, 9, 8, 12]. We start with the best partitioning and then do the worst one and then keep on going in alternating manner. (To select the best or worst pivot, swap your selected pivot with the end (A[r]) before starting the partition)

Best = median

worst = Max / min



Name:

ID:

CSCI 3104, Algorithms
Quiz 3' : 15 points total

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

3. (3 pts) Using the input array A = [6, 7, 5, 11, 12, 2, 14, 3, 10, 9] and the QuickSort algorithm, what is the number of times a comparison is made to the element with value 3 over the duration of the Quicksort algorithm? You need to include the call to Partition as well as the recursive calls to Quicksort. Include a brief explanation with your answer. (Assume the partition algorithm always chooses the last element as the pivot.)

6 | 7 | 11 | 12 | 2 | 14 | 3 | 10 | 9

1x

6 7 2 3

3 times follows. +1 for passing its index to the recursive call?.

There will be 5 comparisons made with this input

- one call for its comparison in the beginning of the partition call when it was previously a pivot
- four more comparisons are made while the quicksort algorithm is sorting values near the value three.

Name:

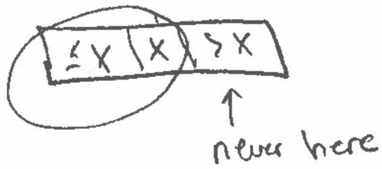
ID:

CSCI 3104, Algorithms
Quiz 3' : 15 points total

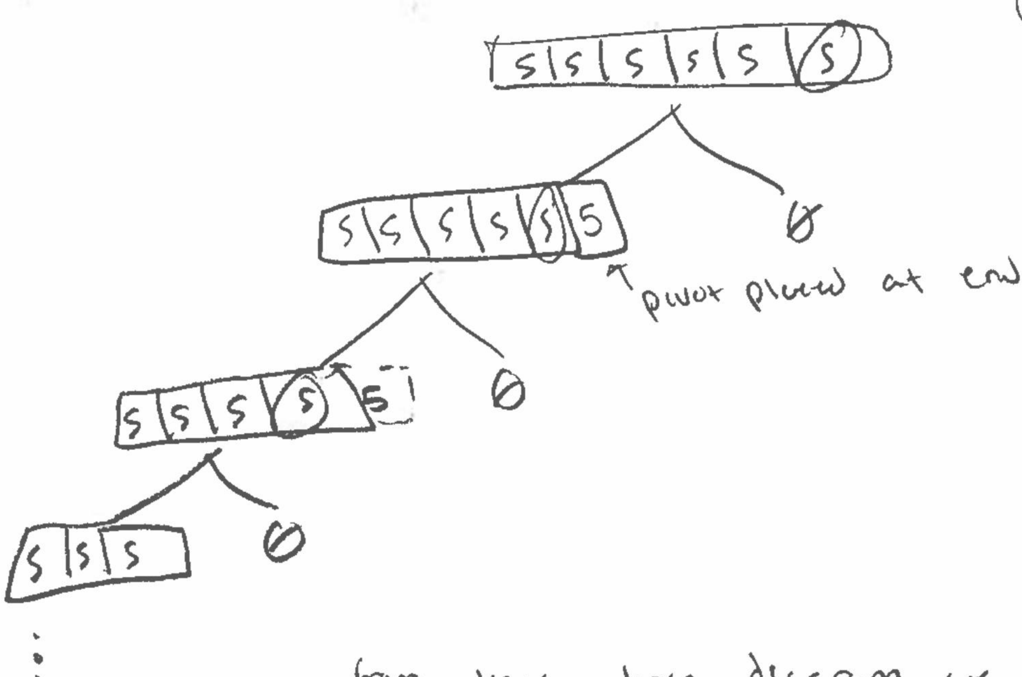
Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

4. (3 pts) Assume you have an array made up of equal elements, such as $A = [5, 5, 5, 5, 5, 5]$ and you call Quicksort to sort the array. Would you see the worst case, best case, or average case performance? Explain what happens in one call to Partition and at least two subsequent calls to Quicksort to illustrate your answer.

You would see the worst case behavior with this array because on the surface level we can examine that there will be awful partitioning done. It will always be a 5:1 split ratio at each recursive call.



EX)



⋮
so on.

from this tree diagram we see $T(n)=T(n-1)+n$ and this ultimately results in $O(n^2)$ run-time which is the worst case for quicksort

Name: ID:

CSCI 3104, Algorithms
Quiz 3' : 15 points total

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

5. (2 pts) Suppose we have a $\mathcal{O}(n^3)$ time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort. Write the recurrence and solve it.

cost	time
findMedian	n^3
partition	n
Quicksort	$2T(n/2)$

← Best case partitioning

← 2 recursive calls that result in same size since best partitioning

$AT(\frac{n}{b}) + \Theta(n)$

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

apply master method

$\log_2 2 = 1 \quad c = 3$

$\log_b a < c \quad \Rightarrow T(n) \in \Theta(n^3)$