CS 101 Fall 2021 - Quiz 3-B

10/19/2021 - 25 Minutes **Name:** **ID number:**

(8 Points) Problem 1: **True or False:** For each statement, write "T" if this statement is correct; write "F" otherwise. Please write your answers in the box below.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
F	T	F	T	F	T	F	F

(1) Quick sort can be implemented in $O(1)$ extra space complexity.

(2) In quick sort (with n distinct elements and sort in ascending order, $n \geq 2$), if we randomly select the pivot, after the first partition operation, the smallest element of the array can be anywhere except the last position.

(3) Merge sort algorithm will have $\Theta(n \log n)$ space complexity in the worst case.

(4) Merge sort never compares the same two elements twice.

(5) Merge sort runs in best case $\Theta(\log n)$ time for certain inputs.

(6) Randomly shuffling the list can reduce the probability of quick sort taking the worst case running time.

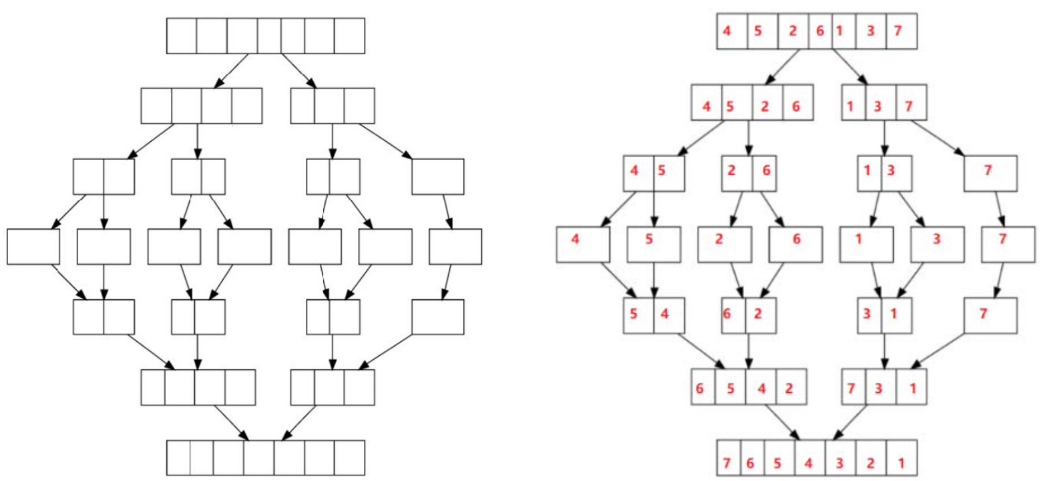
(7) Quick sort has a worst case runtime that is asymptotically better than merge sort's worst case runtime.

(8) Quick sort (ascending) would run the fastest on an already sorted list (ascending) among all sorting methods.

(6 Points) Problem 2: Consider this array: 4, 5, 2, 6, 1, 3, 7.

(1) (4 pts) Use **mergesort** to sort this array in **descending** order. Show your process in the following figure.

(2) (2 pts) How many inversions (if we want to sort in descending order) are there in the array? 9 or 12



(3 Points) Problem 3:

Tom wants to sort his favorite colors in ascending order using quicksort. The original array is:

$$red, cyan, yellow, gray, green, black, blue, white$$

After the first partitioning step, it becomes: ("red" is chosen as pivot)

$$cyan, green, gray, white, black, red, yellow, blue$$

Known that NO elements are equal, we can infer that: (Fill the blanks with ">", "<", or "?" if given informationis insufficient to judge)

(a) red____white (b) yellow____gray (c) green____black

$>, >, ?$

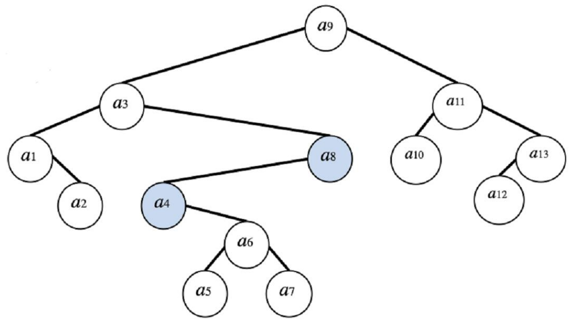
CS 101 Fall 2021 - Quiz 3-B

10/19/2021 - 25 Minutes **Name:** **ID number:**

(8 Points) Problem 4:

(1)(3 pts) Given an array of 8 distinct elements $a_1 < a_2 < ... < a_8$, what is the probability that a_1 and a_8 are compared during **randomized** quicksort?

Hint: Consider BST representation of pivot elements. a_i and a_j are compared once if one is an ancestor of the other.



ans: $\frac{1}{4}$

$\Pr[a_1 \text{ and } a_8 \text{ compared}] = \frac{2}{8}$, they are compared if either a_1 or a_8 is chosen as pivot before any of $\{a_2, a_3, a_4, a_5, a_6, a_7\}$

(2)(3 pts) Given an array of $n(>= 8)$ distinct elements $a_1 < a_2 < ... < a_n$, what is the probability that a_i and a_j are compared during **randomized** quicksort? Briefly explain your answer.

ans: $\frac{1}{4}$

$\Pr[a_1 \text{ and } a_8 \text{ compared}] = \frac{2}{8}$, they are compared if either a_1 or a_8 is chosen as pivot before any of $\{a_2, a_3, a_4, a_5, a_6, a_7\}$

(2)(2 Point) **Analyze** the **expected** number of comparisons during **randomized** quicksort with brief explanation. Use big O notation to express your answer.

Consider BST representation of pivot elements.

- a_i and a_j are compared once if one is an ancestor of the other.
- $\Pr[a_i$ and a_j are compared$] = 2$ / (j-i+1), where $i < j$.

Expected number of compares $= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = 2 \sum_{i=1}^{n} \sum_{j=2}^{n-i+1} \frac{1}{j} \leq 2n \sum_{j=1}^{n} \frac{1}{j} \leq 2n(lnn + 1)$

So the answer is $O(n \log n)$

Answer without explanation can only get 1 points.