CS 101 Fall 2024 - Quiz 3

October, 21, 2024 - 25 Minutes **Name:** **Student ID:**

HINT: $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$

1. (3 points) Honor Code

I promise that I will complete this quiz independently and will not use any electronic products or paper-based materials during the quiz, nor will I communicate with other students during this quiz.

I will not violate the Honor Code during this quiz. ◯ True ◯ False

2. (5 points) True or False

Determine whether the following statements are true or false.

(a) (1') Quick-sort is an in-place sorting algorithm, while merge-sort is not since it requires $\omega(n)$ extra space. ◯ True √ False

(b) (1') Randomized Quick-sort is unstable, while Quick-sort with deterministic method (choosing pivots) is stable. ◯ True √ False

(c) (1') There exists an comparison-based sort algorithm that needs $O(1)$ extra space and takes $o(n \log n)$ time. ◯ True √ False

(d) (1') For an array $\{a_n\}$ with distinct elements, for fixed i, j, if $\forall a_k \neq a_i, a_k \neq a_j, (a_k - a_i)(a_k - a_j) > 0$, then a_i and a_j will be compared in any case when using randomized quick-sort to make $\{a_n\}$ sorted. √ True ◯ False

(e) (1') When we use divide and conquer to solve a problem, we should divide the problem into one or more subproblems with the exact same scale, then recursively do them and merge their answers at last. ◯ True √ False

> **Solution:** Notice that many divide-and-conquer algorithms divide the problem into subproblems of different scales. (e.g. quick sort, median of median)

3. (5 points) Randomized quick-sort

(a) (1') If we use randomized quick-sort(i.e. randomly choosing pivots) to sort the array $[3, 4, 6, 2, 1, 5, 8, 0]$, the probability of 1 and 6 are compared is $\frac{1}{3}$.

(b) (1') Use the same method as above to sort an array with n distinct elements, the probability of i-th largest and j-th largest element ($i \neq j$) are compared is $\frac{2}{|j-i|+1}$.

(c) (3') Prove that the expectation times of comparisons in the randomized quick-sort is $\Theta(n \log n)$.
Hint: The total expectation times can be obtained from the sum of the expectation of each comparison.

Solution: Denote the expectation time as E, $p_{i,j}$ as the probability that $i-$th element and $j-$th element are compared:

$$\begin{aligned}
E &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} p_{i,j} = \sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= 2\sum_{i=1}^{n}\sum_{k=2}^{n-i+1} \frac{1}{k} = 2\sum_{k=2}^{n}\sum_{i=1}^{n-k+1} \frac{1}{k} \\
&= 2\sum_{k=2}^{n} \frac{n-k+1}{k} = 2(n+1)\sum_{k=2}^{n}\frac{1}{k} - 2(n-1) \\
&= 2(n+1)\Theta(\log n) - 2(n-1) = \Theta(n\log n).
\end{aligned}$$

4. (12 points) Solving Recursion

Solve the recursion relation with $T(1) = 1, T(0) = 0$:

(a) (3') $T(n) = T(n-1) + \Theta(c^n)$ ($c > 1$ is a constant).

Solution:

$$T(n) = T(n-1) + c^n = \sum_{i=2}^{n} c^i + 1 = 1 + c^2 \frac{c^{n-1} - 1}{c - 1} = \Theta(c^n)$$

(b) (4') $T(n) = T(\frac{n}{2}) + \Theta(\log n)$

Solution: Denote $k = \log_2 n$, $f(k) = T(n)$, then $f(0) = T(1) = 1$

$$\begin{aligned} f(k) = T(n) &= T(\frac{n}{2}) + \Theta(\log n) \\ &= f(k-1) + \Theta(k) \\ &= \sum_{i=1}^{k} \Theta(i) = \Theta(k^2) \end{aligned}$$

So $T(n) = f(k) = \Theta(k^2) = \Theta(\log^2 n)$

(c) (5') $T(n) = \Theta(n) + \frac{1}{n}\sum_{i=0}^{n-1}(T(i) + T(n-i))$, you can write $\Theta(n)$ as cn for your convenience.

Solution:

$$\begin{aligned} T(n) &= cn + \frac{1}{n}\sum_{i=0}^{n-1}(T(i) + T(n-i)) \\ \frac{n-1}{n}T(n) &= cn + \frac{2}{n}\sum_{i=1}^{n-1} T(i) \\ (n-1)T(n) &= cn^2 + 2\sum_{i=1}^{n-1} T(i) \end{aligned}$$

And trivially: $nT(n+1) = c(n+1)^2 + 2\sum_{i=1}^{n} T(i)$, then:

$$\begin{aligned} nT(n+1) - (n-1)T(n) &= c(2n+1) + 2T(n) \\ \frac{T(n+1)}{n+1} - \frac{T(n)}{n} &= c\frac{2n+1}{n(n+1)} \\ \frac{T(n+1)}{n+1} &= 1 + c\sum_{i=1}^{n}(\frac{1}{i} + \frac{1}{i+1}) \\ \frac{T(n+1)}{n+1} &= 1 + c(\sum_{i=1}^{n}\frac{2}{i} + \frac{1}{n+1} - 1) = 1 + \Theta(\log n) \\ T(n+1) = \Theta(n \log n) &\Rightarrow T(n) = \Theta(n \log n) \end{aligned}$$