

CSE310 HW01, Tuesday, 02/02/2021, Due: Tuesday, 02/09/2021

Please read the instructions carefully. **You have to use the companion answer sheet (which is a fillable PDF file) to type/select your answers to the questions described here.** Adobe Acrobat Reader can be found at <https://get.adobe.com/reader/>. Hand-written assignment (or photo of it) will not be graded. **Submit the filled PDF file of the answer sheet on Gradescope, following the link on Canvas.** You should name your file using the format **CSE310-HW02-LastName-FirstName.pdf**.

Q1 (4 points) We have the recurrence relation $T(n) = 8 \cdot T(n/2) + n^3$. Suppose that we want to use the Master method to solve $T(n)$. We need to decide the case and derive the conclusion.

- (a) On the answer sheet, decide which case of the Master method this recurrence falls into, by checking the corresponding box.
- (b) On the answer sheet, decide the asymptotic notation of $T(n)$ by checking the corresponding box.

Q2 (10 points) There are two algorithms A and B for solving the same problem. Algorithm A has a worst-case time complexity $T_A(n)$ that satisfies the recurrence relation

$$T_A(n) = 7 \cdot T_A(n/2) + n^3 \log n,$$

where n is the input size. Algorithm B has a worst-case time complexity $T_B(n)$ that satisfies the recurrence relation

$$T_B(n) = 8 \cdot T_B(n/2) + n^2,$$

where n is the input size. Suppose that you want to use the Master method to find the asymptotic notations for $T_A(n)$ and $T_B(n)$, and to decide which algorithm is faster.

- (a) On the answer sheet, decide which case of the Master method $T_A(n)$ falls into, by checking the corresponding box.
- (b) On the answer sheet, decide the asymptotic notation of $T_A(n)$ by checking the corresponding box.
- (c) On the answer sheet, decide which case of the Master method $T_B(n)$ falls into, by checking the corresponding box.
- (d) On the answer sheet, decide the asymptotic notation of $T_B(n)$ by checking the corresponding box.
- (e) On the answer sheet, decide which algorithm is faster by checking the corresponding box.

Q3 (4 points) This question tests your understanding of activation records as taught in class.

- (a) Assume that we want to apply Quicksort to sort the array A with 5 elements where the initial values of the array elements (from $A[1]$ to $A[5]$) are $A : \boxed{19 \mid 16 \mid 13 \mid 25 \mid 22}$. How many times do we push an activation record (for quicksort or partition) onto the runtime stack before the completion of the algorithm? Note that `Quicksort(A, 1, 5)` is the first activation record pushed onto the runtime stack.
- (b) Assume that we want to apply mergesort to sort the array A with 5 elements where the initial values of the array elements (from $A[1]$ to $A[5]$) are $A : \boxed{19 \mid 16 \mid 13 \mid 25 \mid 22}$. How many times do we push an activation record (for mergesort or merge) onto the runtime stack before the completion of the algorithm? Note that `mergesort(A, 1, 5)` is the first activation record pushed onto the runtime stack.

Q4 (12 points) Assume that we use Quicksort (as in the textbook) to sort the array A with 5 elements where the initial values of the array elements (from $A[1]$ to $A[5]$) are $A : \boxed{19 \mid 16 \mid 13 \mid 25 \mid 22}$. This question tests your understanding of the algorithm by following the element-wise comparisons made in the execution of the algorithm. Here an element-wise comparison means the comparison of one element of the array with another element of the array or the key set in a particular step of the algorithm (Line 4 in Partition). Since the algorithm may move the elements of the array, you need to show the values of the elements being compared (rather than the form of $A[i]$). The first element-wise comparison is

$$19 \leq 22?$$

On the answer sheet, answer the following questions by checking the corresponding box.

- (a) What is the 2nd element-wise comparison?
- (b) What is the 4th element-wise comparison?
- (c) What is the 6th element-wise comparison?
- (d) What is the 7th element-wise comparison?

In order to answer the above questions correctly, you need to execute the algorithm on the given input step by step.

Q5 (10 points) On your scratch paper, draw the portion of the decision tree for insertion sort on 5 elements a_1, a_2, a_3, a_4, a_5 that contains the path from the root node to the permutation $\langle a_3, a_2, a_1, a_5, a_4 \rangle$. Then answer the following questions on the answer sheet. Here the 1st node on the root to leaf ($\langle a_3, a_2, a_1, a_5, a_4 \rangle$) path is the root node, which contains the comparison $\boxed{a_1 > a_2 ?}$.

- (a) What is comparison at the 2nd node on the root to leaf path?
- (b) What is comparison at the 3rd node on the root to leaf path?
- (c) What is comparison at the 4th node on the root to leaf path?
- (d) What is comparison at the 5th node on the root to leaf path?
- (e) What is comparison at the 6th node on the root to leaf path?

Q6 (12 pts) This question tests your understanding of the time complexities of sorting algorithms.

- (a) In the decision tree for insertion sort on n elements, what is the length of the shortest root-to-leaf path? Check the corresponding box on the answer sheet.
- (b) In the decision tree for insertion sort on n elements, what is the length of the longest root-to-leaf path? Check the corresponding box on the answer sheet.
- (c) In the decision tree for quicksort on n elements, what is the length of the shortest root-to-leaf path? Check the corresponding box on the answer sheet.
- (d) In the decision tree for quicksort on n elements, what is the length of the longest root-to-leaf path? Check the corresponding box on the answer sheet.
- (e) In the decision tree for mergesort on n elements, what is the length of the shortest root-to-leaf path? Check the corresponding box on the answer sheet.
- (f) In the decision tree for mergesort on n elements, what is the length of the longest root-to-leaf path? Check the corresponding box on the answer sheet.