Revised 09/22/2022

A. Y. 2022 – 2023

FIRST AND SECOND SEMESTER

PRACTICE PREMIDTERM EXAM 2

CS 3102N – Algorithms and Complexity

Name:	Christian Stewart
Program and Year:	BSCS 3rd Year
CS 3102N Group #:	1
Date:	27/09/22

NOTES:

I. You have **120 minutes** to take the test. Kindly inspect the exam time duration carefully and do not spend too much time on a single question.

II. Each test has different directions. Follow them carefully.

III. Answer each item by typing your answers in the fields and spaces provided.

IV. Always try to write accurately using appropriate and efficient program logic and proper coding syntax. For programming problems, follow the coding conventional rules that are set for the class (e.g. all conditional statements must involve relational operators; break and continue statements shall be only used in switch blocks.).

V. Usage of compilers, calculators, and/or related applications is **STRICTLY NOT ALLOWED**.

VI. Should you have a concern on any of the questions, kindly contact or message the examiner for inspection or clarification.

VII. Answer each question as far as you can. Try not to leave blanks.

VIII. Once you are finished with the exam, save the PDF file with your answers and submit it as an attachment to the examiner's email address (20100215@usc.edu.ph) or through direct message for checking.

I. MULTIPLE CHOICE

DIRECTIONS: Choose the letter of the correct of best answer. Each item is worth 1 point.

A ..... 1. If a heap is implemented to form a full binary tree with height 9 when the heap is fully populated and elements are populated starting at index 0, what must be the size of the array?
[A] 1023 [B] 1024 [C] 511 [D] 512 [E] None of the choices

C ..... 2. In a max-heap, the element with the greatest key is always in which node?
[A] Any leaf node [B] First node of left subtree [C] Root node [D] First node of right subtree
[E] Leftmost leaf [F] Rightmost leaf [G] None of the choices

D ..... 3. Consider a heap with 250 active elements and the root at index 0. If a heapify procedure is to be implemented, at what index of the array should the process start with?
[A] 249 [B] 0 [C] 125 [D] 124 [E] 123

A ..... 4. A binary min heap, implemented using array, has 100 nodes. The root is stored at index 1. Which of the following statements is FALSE?
[A] The POT has at least 37 leaves.
[B] The POT has a maximum path of length 6.
[C] The element at index 1 must be less than or equal to that at index 2^n for all positive integers n.
[D] The last occupied element is at index 100.
[E] None of the choices

C ..... 5. A complete binary min-heap is made by including each integer in [1, 1023] exactly once. The depth of a node in the heap is the length of the path from the root of the heap to that node. Thus, the root is at depth 0. The maximum depth at which the integer 9 can appear is:
[A] 3 [B] 7 [C] 8 [D] 9 [E] None of the choices

B ..... 6. Which of the following is not a characteristic of a partially ordered tree?
[A] The height is always the minimum possible for the current number of nodes.
[B] At the lowest level, all the missing nodes are to the left of the nodes present at the lowest level.
[C] The priority of the parent is less than or equal to that of its children.
[D] Each node of the tree cannot exceed 2 immediate children.

B ..... 7. Which of the following is not a stable sorting algorithm in its typical implementation.
[A] Insertion sort [B] Quick sort [C] Merge sort [D] Bubble sort [E] None of the choices

C ..... 8. Among the following algorithms, which can be considered the most efficient for sorting a list of elements where only a few of them are left at their unsorted positions?
[A] Strand sort [B] Quick sort [C] Merge sort [D] Heap sort [E] Insertion sort

A ..... 9. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: **2 5 1 7 9 12 11 10**
Which statement is correct?
[A] The pivot could be either the 7 or the 9. [B] The pivot could be the 7, but it is not the 9
[C] The pivot is not the 7, but it could be the 9 [D] Neither the 7 nor the 9 is the pivot.
[E] The pivot could not be detected

D ..... 10. Quicksort performs best when the pivots selected in the recursive calls are:
[A] the mean of the given set [B] the last position of the given set [C] the first position of the given set
[D] the median of the given set [E] the middle position of the given set [F] None of the choices

E ..... 11. Which among the following pairs of sorting algorithm and their proponents is incorrect?
[A] Merge sort - John von Neumann [B] Quicksort - Charles Antony Richard Hoare
[C] Heapsort - J. W. J. Williams [D] Tournament sort - Alexander Stepanov and Aaron Kershenbaum
[E] None of the choices

II. STRUCTURED RESPONSE

DIRECTIONS: Read carefully and answer the questions correctly. To gain full marks to questions you should express your ideas sensibly and answer with the proper syntax.

1. Complete the table by filling in the correct information for the following algorithms.
   - For complexity analysis, Use **n** for number of elements and **k** for the range of the input.
   - For characteristics, consider the following: **S**-stable, **U**-unstable, **CA**-comparison approach, **NCA-**non-comparison approach, **IP**-in-place, **OP**-out-of-place.
   - Indicate the year of discovery for each algorithm.

Name	Time Complexity			Space Complexity	Characteristics	Year Discovered
	Best	Average	Worst			
Heapsort	O(nlogn)	O(nlogn)	O(nlogn)	O(1)	U,CA,IP	1964
Tournament Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)	U,CA,OP	2002
Merge sort (Abs. in Place)	)(nlogn^2	(n logn^	O(n^2)	O(1)	S,CA,IP	1945
Merge sort (Bottom up)	O(nlogn)	O(nlogn)	O(nlogn)	O(n)	S,CA,OP	1945
Merge sort (Top down)	O(nlogn)	O(nlogn)	O(nlogn)	O(n)	S,CA,OP	1945
Quicksort	O(nlogn)	O(nlogn)	O(n^2)	O(1)	U,CA,IP	1960
Strand sort	O(n)	O(nlogn)	O(n^2)	O(n)	S,CA,OP	----

[24]

2. Complete the statements below with the appropriate words or phrases.

   (a) We can say that an algorithm is **adaptive** if it takes advantage of the position of its elements to minimize the number of operations needed to perform the sort.

   (b) **strand** sort is a recursive algorithm that repeatedly pulls out a series of elements from the unsorted list into a sub list to be sorted, then merges them into the result list, and is derived from **selection** sort.

   (c) A **divide and conquer** algorithm is a strategy of solving a large problem by breaking the problem into smaller sub-problems, solving the sub-problems, and finally combining them to get the desired output to use the said algorithm, a concept called **recursion** usually has to be applied.

   (d) **tournament** sort is a generalization of **binomial** trees of Brown and Vuillemin. It is a variation of heap sort but requires an auxiliary storage as much as **twice** the size of the original list.

   (e) The recursive calls done in the merge sort and quicksort algorithms do resemble a pattern similar to that of traversing a **binary tree** , in which quicksort does it in a/an **preorder** type of traversal while merge sort does it in a/an **postorder** type of traversal.

[10]

3. (a) A max heap is implemented in an array whose root is located at index 0.

(i) The right child of index 26 is at index **54** **[1]**

(iii) The parent of index 74 is at index **36** **[1]**

(iv) Within indices 41-60 of the heap, the element at index 5 must be always larger than or equal to the elements in what indices? **47-54** **[2]**

(b) A min heap is implemented in an array whose root is located at index 1.

(i) The right child of index 9 is at index **19** **[1]**

(iii) The ancestors of index 30 excluding itself are at indices **15,7,3,1** **[1]**

(iv) If a new element is initially inserted at index 67, the possible final positions of this element are at indices **67,33,16,8,4,2,1** **[2]**

4. Christine, a DSA enthusiast, owns a lot of cars after working for 20 years in CCLEX – *sana all :((*. She has a very large garage at home, so she wants to do some experiments with her cars by forming a heap and sort out her cars according to plate numbers.

SEN 3023, KIR 1363, NAM 2104, KON 0046, KYO 2005,
SUM 0406, ARI 1747, DOI 4947, KAN 9537, WAK 1006

Note: AAA 0000 < ZZZ 9999

(a) Using those values, build the **min-heap** by inserting the given strings in order into an initially empty POT Complete the table by identifying the strings present in the selected array indices.

Scenario	**A[0]**	**A[3]**	**A[5]**
Initial heap	**ARI 1747**	**KAN 9537**	**SUM 0406**
After removing two (2) smallest elements	**WAK 1006**	**SEN 3023**	**SUM 0406**

[6]

(b) Assuming the given strings appear in the exact order in an array, build the **max-heap** by heapifying starting from the lowest level parent. Complete the table by identifying the strings present in the selected array indices.

Scenario	**A[0]**	**A[3]**	**A[5]**
Initial heap	**WAK 1006**	**KIR 1363**	**NAM 2104**
After removing two (2) largest elements	**SEN 3023**	**KIR 1363**	**KAN 9537**

[6]

5. Use the characters of the string **"volkswagen"** to perform the indicated sorting algorithms below. Provide the:

a) Strand sort → output string after the 4th recursive call. **klnosvw** **[5]**

b) Quick sort (Hoare's variation with pivot at leftmost element)
→ output string at the end of the third partition call: **aelksnogvw** **[4]**

c) Quick sort (Hoare's variation with pivot at rightmost element, median-of-three technique applied)
→ output string at the end of the third partition call: **aelkngosvw** **[4]**

d) Merge sort (bottom-up) → output string after the second outer loop iteration: **klovagswen** **[4]**

e) Merge sort (top-down) → output string after the fifth merge call: **klosvawgen** **[4]**

6. Given the following list of values: **8, 22, 7, 9, 11, 5, 13, 20**

   a) Perform tournament sort to arrange elements in increasing order and list the contents of the auxiliary array:

      i) When the first sorted element is detected: **13,10,13,8,10,11,13,15,8,22,7,9,11,INF,13,20** **[2]**

      ii) When the third sorted element is detected: **8,8,11,8,9,11,14,15,INF,22,INF,11,INF,13,20** **[2]**

   b) Perform tournament sort to arrange elements in descending order and list the contents of the auxiliary array:

      i) When the first sorted element is detected: **9,9,14,15,9,12,14,15,8,INF,7,9,11,5,13,20** **[2]**

      ii) When the third sorted element is detected: **14,8,14,8,7,12,14,INF,8,INF,7,9,11,5,INF,INF** **[2]**

7. Identify and compare between the variations of each sorting algorithm in terms of their implementation and running time and space usage efficiency. Determine which among the variations do you think is the most efficient to implement.

   a) Heapsort and tournament sort

      Space complexity of heapsort is O(1) while tournament sort is O(n).

      [4]

   b) Merge sort – abstract in-place, bottom-up approach, top-down approach

      Abstract in-place has better space implementation of O(1) but its time complexity increases dramatically with a worst case of O(n^2) compared to the bottom-up and top-down worst case of O(nlogn). Bottom-up however, is better compared to top-down since bottom-up uses an iterative method which saves space compared to top-down which is uses another array on top of another array because of recursion

      [4]

   c) Quick sort – Hoare's variation, Lomuto's variation

      [3]

8. In Hoare's variation of quicksort, it is said that the pivot's final location is not necessarily at the index that was returned, and the next two segments that the main algorithm recurs on are (lo..p) and (p+1..hi) as opposed to (lo..p-1) and (p+1..hi) as in Lomuto's scheme. Why is this so?

   [2]

9. In the discussion, it was mentioned that recursive algorithms/solutions, although efficient in approach, is not optimal to use all the time/in all scenarios. Do you agree? Support your answer with various examples.

   [5]

10. An advantage of tournament sort is that the algorithm can efficiently find the "next best winner" in the tournament immediately after the "winner" has been determined. Explain how and give the time complexity in doing so.

    After finding the winner, the algorithm goes through another iteration by changing the winner of the previous tournament to a sentinel value and continue with comparisons and swaps accordingly. This allows the time complexity of determining the next best winner to be O(nlog n - 2).

    [3]

11. a) Does strand sort have a better performance when implemented entirely using array or linked list? Explain.

Strand sort is better implemented on linked list compared to an array because it requires many insertions and deletions which can be O(1) on a linked list.

[3]

b) If we implement strand sort by using a stack instead of a queue as a sub list/sorted list to gather the elements prior to merging with the output list, in what manner must the element be gathered to retain the algorithm's efficiency in the original variation/presentation where a queue is used for the sub list?

[4]

12. On what scenarios does each of the sorting algorithms below perform at its worst?

a) Heapsort	**Heapsort is a non-adaptive sorting algorithm. It has O(nlogn) for all cases**
b) Tournament sort	**Tournament sort is a non-adaptive sorting algorithm. It has O(nlogn) for all cases**
c) Quicksort	**Quicksort performs worst when the pivot chosen is either the least or greatest element. It has O(n^2)**
d) Merge sort	**Merge sort is a non-adaptive sorting algorithm. It has O(nlogn) for all cases**
e) Strand sort	**Strand sort performs worst when elements are sorted in reverse order. It has O(n^2)**

[10]

III. PROGRAMMING

DIRECTIONS: Read carefully and give accurately what is asked. Make your code concise, efficient, and readable. All functions should have up to one (1) return statement only.

13. Given the data structure on the right and the idea that the value of stored in index 0 represents the number of elements found in the list, write the code of function **isMaxHeap()**. The function checks if the given array represents a max heap and returns TRUE if so and FALSE if it isn't.

```
//Definition
#define MAX 101
typedef unsigned int SET[MAX];
typedef enum {TRUE,FALSE} Boolean;
```

[10]

```
typedef struct{
 int elem[SIZE];
 int lastNdx;
 int heapLastNdx;
}Heap;

int isMaxHeap(Heap H){
 int P,SC,isHeap=1,last=(H.lastNdx-1)/2;
 for(P=0;P<=last && isHeap==1;P++){
  isHeap=1;
  if(P*2+2 > H.lastNdx){
   SC = P*2+1;
  } else if (P*2+1 <= H.lastNdx){
   SC = (H.elem[P*2+1] < H.elem[P*2+2]) ? P*2+1 : P*2+2;
  } else {
   SC = P;
  }
  if(H.elem[P]<H.elem[SC]){
   isHeap=0;
  }
 }
 return isHeap;
}
```

14. Implement Quick sort (Hoare's variation) using a doubly linked list data structure defined on the right.

    Given a list of elements in random order, these are to be sorted in increasing order of the values in the data field. Use the leftmost element as the pivot element.

    Ensure all dynamically allocated memory is retuned accordingly to avoid any memory leaks. Declare your own auxiliary data structures as necessary.

```
//Definition
typedef struct cell1{
   int data;
   struct cell1* prev;
   struct cell1* next;
}node1, *List1;
```

[24]

15. Implement Merge Sort (top-down variation). using a singly linked list data structure defined on the right.

    Given a list of elements in random order, these are to be sorted in increasing order of the values in the data field by applying the

    Ensure all dynamically allocated memory are retuned accordingly to avoid any memory leaks. Declare your own auxiliary data structures as necessary.

```
//Definition

typedef struct cell2{
   int data;
   struct cell2* next;
}node2, *List2;
```

[24]

"If you're always trying to be **normal**, you will **never know** how **amazing** you can be."
- Maya Angelou -

=== THE END ===

God bless you!

REVIEW YOUR ANSWERS!

Prepared by:
Wayne Dayata