CS2040 2425 Sem1 Mock Midterm

[Solutions]

Your Name: ______________________________ **Your ID:** ______________________________

# of Questions: 8

Date and Time of Exam Creation: Mon, Sep 23, 2024 @ 15:08:55

Total Exam Points: 28.00

Question #: 1

Which of the following statements are **false** about sorting?
[3 marks]

(Select all options you think are **false**.)

A. Merge sort and Quick sort are stable sorts.
B. Improved Bubble sort (stop once a pass through the array does not cause any swapping) can run faster than O(n^2) in the worst case.
C. Comparison based sorting cannot be faster than O(n lg n) in the worst case.
D. Radix sort of an integer array has time complexity O(n) when the maximum number of digits for any integer value in the array is bounded by some constant c.

Item Weight: 3.0

Question #: 2

You are given a queue Q that contains some integers and a helper ADT H that is empty. You are now asked to search and find out whether an integer k is in Q. You are allowed to move the data between Q and H using standard ADT operations, but you are not allowed to store the data anywhere else (not even in temporary variables). After searching, you must restore Q and H to their original state. In other words, Q should contain all the integers in the original order, whereas H must be empty. Which of the following options for H can help us achieve our goal?
[3 marks]

(Select all options you think can help us achieve our goal.)

A. H is a linked list ADT.
B. H is a stack ADT.
C. H is a queue ADT.
D. None of the above.

Item Weight: 3.0

Question #: 3

Which of the following linked lists can be used to implement a Queue data structure in a manner that guarantees Poll (dequeue) and Offer (enqueue) operations can be performed in O(1) worst case time complexity?
[3 marks]

(Select all options you think are correct.)

A. A tailed linked list
B. A doubly linked list (with both head and tail reference)
C. A circular linked list (i.e., a tailed linked list where tail points to head)
D. None of the above are possible

Item Weight: 3.0

Question #: 4

Suppose you have a hash table of size 8, and you want to design a hash function for integer keys. Among the following functions, which one would be the best?
[3 marks]

A. h(key) = 8
B. h(key) = (12 * key + 4) % 8
C. h(key) = (7 * key + 9) % 8
D. h(key) = (key % 5) % 8

Item Weight: 3.0

Question #: 5

Questions 5 and 6 refer to the statement given below.
The time complexity of the following program when ***n* >O(*m*^{2})** is O(*n***m*).
[2 marks]

```
public static void main(String[] args){
  Scanner sc =new Scanner(System.in);
  int n =sc.nextInt();
  int m =sc.nextInt();
  int result =0;
  for (int i=0; i <n; i++) {
    for (int j=m; j >0; j--) {
      result +=1;
    }
    m -= 1;
  }
  System.out.println(result);
}
```

A. True

B. False

Item Weight: 2.0

Question #: 6

The reason for your answer for Q5 is because:
[2 marks]

A. The outer loop will iterate *n* times and inner loop iterates *m* tines, so in total the time complexity is O(n*m).

B. Since *n* >O(m^2) the time complexity should be O(m^3) and not O(n*m).

C. The inner loop stops when *m* reaches 0, so the statement in the inner loop is executed O(m^2) times, but the outer loop will still keep on running, since *n* >O(m^2). Thus the time complexity is O(n) and not O(n*m).

Item Weight: 2.0

Question #: 7

Questions 7 and 8 refer to the problem described below.
Suppose you are given a very large 0-based integer array A with *m* (*m* >10) elements, with elements sorted in descending order. However, only the first *n* (*n* <*m*) elements contain data which are positive integers, and the rest of the array elements contain zeroes. The values of *n* and *m* are given to you.

Select the correct and most efficient algorithm (with the correct time complexity given) among the options given below for a method **search(A,*k*)** to search for a key *k* in the array A. It returns the index of the array where *k* is found, or -1 otherwise.
[5 marks]

A. Perform a linear search on A from index 0 to index m-1 to find k. This will take worst case

O(m) time.

B. Perform a binary search on A for k starting with the entire array (i.e., binary search between index 0 and index m-1, inclusive of 0 and m-1). This will take worst case O(log m) time.

C. Perform a binary search on A for k starting with the entire array (i.e., binary search between index 0 and index m-1, inclusive of 0 and m-1). Since there are only 1's and 0's in the array the worst case time complexity should be m).

D.
```
if (k != 0 and k != 1)
    return -1
else
    if (k == 0 and n > 0)  return 0 else return -1
    if (k == 1 and n return n else return -1
```

This takes worst case O(1) time.

E. Perform a binary search on A for *k* starting with a partial array (i.e., binary search between index 0 and index *n*-1, inclusive of 0 and *n*-1). This will take worst case O(log *n*) time.

F. Perform a linear search on A starting from index 0 to index n to find k. This will take worst case O(n) which is less than O(m).

Item Weight: 5.0

Question #: 8

If *n* is not given (i.e., the number of positive integers in the array is not given), give the best algorithm you can think of to solve the problem. **[7 marks]**

Perform a binary search on A for k starting with the entire array (i.e., binary search between index 0 and index m-1, inclusive of 0 and m-1). This will take worst case O(log m) time.

Item Weight: 7.0