CS203 Data Structure and Algorithm Analysis   Mid-term 2021Fall

Student ID:__________   Student Name:____________

Lab time:☐ Tue.3-4   ☐ Tue.5-6   ☐ Wed.5-6   ☐ Wed.7-8

Part I. Filling-blank question [30 marks, 3 marks for each question]

1. The enqueue sequence of a queue is 4 3 2 1. Which of the following dequeue sequence is /are possible? ______

   A. 1 2 3 4   B. 4 3 2 1   C. 4 1 3 2   D. 3 2 4 1

2. Given an ordered sequence with 16 elements, the binary search algorithm is used to find a target element (the element is not in the sequence), then the comparison times between the element corresponding to mid and the target element are at least ______?

   A. 4   B. 5   C. 6   D. 7   E.None of above

3. (True or False)The storage addresses of each node in the linked list are continuous. ______

4. What is the time complexity of the following method ______.

```
method(n){
   x=0;
   while ( (x+1)*(x+1) <= n )
      x++;
}
```

5. Given an array A with size m, suppose there are n (n<m) integers store in A[0], A[1], A[2],⋯, A[n-1]. If we delete A[i] from A[] (i<n), we need to move ______ integers.

6. The original expression for the postfix expression of ab/cd*ef*-g/+ is ___________________.

7. The time complexity of inserting a new node into an ordered single linked list with length N and keeping the linked list still in order is _______.

8. Let the initial state of stack S and queue Q be empty. The push-in sequence of stack S is abcdefg. If each element enqueue Q immediately after leaving the stack S, and the dequeue sequence of Q is bdcfeag, there are at most ______ elements in stack S in this process.

9. Given next array next[], of which next[i](0<=i<=n-1) represents the longest common prefix of P[0,...,i]. Please write down a string whose next array is {0,1,0,1,2,2,3}, only characters {a, b, c} can be used. ________________

10. Suppose that we use an array A [0, … , m] to store the elements of a circular queue. The operation of enqueue is ______________________. (If front == rear, the queue is empty.)

    A. rear=rear+1 B. rear=(rear+1) mode (m-1) C. rear=(rear+1) mod m D. rear=(rear+1) mod (m+1)

Part II. Short answer question [20 marks, 5 marks for each question]

1. Given a search pattern string “abadaabc” (index starts at 0), please finish the following tables of a FSA constructed for string matching:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
a	1		3		5	6			
b		2					7		
c								8	
d				4					

j	Pattern[1...j]	X
0		0
1		
2		
3		
4		
5		
6		
7		

2. Give the structure of node of a single linked list as:

Java:
```
class Node {
    int val;
    Node next;
}
```

C\C++:
```
struct Node{
    int val;
    Node * next;
}
```

Please use the single linked list **Link** to simulate a stack and implement the methods of push and pop. **Link** is a linked list with a head node **Link**. (When stack **Link** is empty, node **Link** is not null, but next of **Link** is null.)

```
void push(int val, Node* Link)
{

}
```

```
int pop(Node* Link){
    int val;

    return val;
}
```

3. Function A is an implementation of binary search algorithm to find the largest index of a target element x (x <= k) in a non-descending size-n array Arr. Please fill in the blanks.(‘;’ and ‘:’ are not allowed in the contents filled in)

```
int A(int Arr[], int k)
{
   int min = 0, max = Arr.length-1, mid;
   while(min ______ max){
       mid = ____________________;
       if (Arr[mid] _______ k){
           min = __________;
       }else{
           max = __________;
       }
   }
   return ___________;
}
```

4. The following array is to be sorted in ascending order: 8,6,1,3,7,2,5,4. The process of sorting the order of the data by Selection Sort and Insertion Sort is:
(Please fill in the blanks)

Selection Sort									Insertion Sort								
Step1	1	6	8	3	7	2	5	4	Step1	8	6	1	3	7	2	5	4
Step2	1								Step2								
Step3	1								Step3	1							
Step4	1								Step4	1							
Step5	1								Step5	1							
Step6	1								Step6	1							
Step7	1								Step7	1							
Step8	1	2	3	4	5	6	7	8	Step8	1	2	3	4	5	6	7	8

Part III. Algorithm [50 marks]

(1) [Divide and Conquer, 13 marks]

The Fibonacci bit-string FS(n) is defined as follows:

FS(0) = "0"

FS(1) = "1"

FS(n) = FS(n−2) ⊕ FS(n−1), if n ≥ 2 The operator ⊕ is used to concatenate two bit-strings.

Here are examples of the bit-string FS(n). Note that the length of FS(n) forms a Fibonacci series.

n	0	1	2	3	4	5	…
FS(n)	"0"	"1"	"01"	"101"	"01101"	"10101101"	…
Length of FS(n)	1	1	2	3	5	8	…

Assume that we count bit positions in a bit-string from left to right. For example, the 1-st, 2-nd, 3-rd, 4-th, 5-th bit of FS(4) are 0, 1, 1, 0, 1, respectively. Since the bit-string FS(9) is built from FS(7) and FS(8), we want to express the k-th bit of FS(9) as either the i-th bit of FS(7) or the j-th bit of FS(8). (where i, j are some integers)

(1) Fill in the following blanks **[2 marks]**:

The 7-th bit of FS(9) must come from the _________-th bit of FS(______).

The 32-th bit of FS(9) must come from the _________-th bit of FS(______).

Now we just want to find the k-th bit of FS(n) efficiently, without generating the entire FS(n).

(2) Write a recursive algorithm to find the k-th bit of FS(n). Your algorithm may use this name: "FindFS(Integer n, Integer k)". **[5 marks]**

(3) Draw the path of the recursive calls used for computing "FindFS(9, 33)". **[3 marks]**

(4) Analyze the worst-case time complexity of your algorithm in Question (2). **[3 marks]**

(2) [Sorting Algorithm, 17 marks]

In insertion sort, we move elements only one position ahead. When an element has to be moved far ahead, many movements are involved. Now, we consider a variation of Insertion Sort: **NewSort**. The idea of **NewSort** is to allow exchange of far items. In **NewSort**, we make the array h-sorted for a large value of h. We keep reducing the value of h until it becomes 1. An array is said to be h-sorted if all sublists of every h'th element is sorted.

(a) [2 marks] Records A[1], A[2], A[3],…, A[N] are said to be h-sorted, if ________

(A) A[i] <= A[i+h] for 1<= i*h <= N

(B) A[h] <= A[i+h] for 1<= i <= N

(C) A[i] <= A[h] for 1<= i <= h

(D) A[i] <= A[i+h] for 1<= i <= N-h

(b) [2 marks] An array that is first 7-sorted, then 5-sorted becomes _________

(A) 7-ordered

(B) 5-ordered

(C) both 2-ordered and 5-ordered

(D) both 7-ordered and 5-ordered

(c) [4 marks] In the worst case, the QuickSort algorithm and **NewSort** algorithm will degenerate to ________ and ________ sort algorithm, respectively.

(d) [3 marks] **NewSort** is more efficient than insertion sort if the length of input arrays is small. True or False? Why?

(e) [6 marks] Fill the following table to show the running steps of **NewSort** Algorithm.

	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
Input	14	25	23	35	6	20	78	66
4-Sorted								
2-Sorted								
1-Sorted								

(3) [Algorithm Design, 20 marks]

Let y^i denote the concatenation of string y with itself i times. For example, (abc)^4 = abcabcabcabc. We say that a string x has repetition factor r if x = y^r for some string y and some r > 0. For example, abcabcabcabc = (abc)^4 or (abcabc)^2. Let p(x) denote the largest r such that x has repetition factor r. p(abcabcabcabc) = 4.

Given a pattern P[1,…,m] and computes the value p(P[1,…,i]) for i =1,…,m.

(a) [6 marks] Fill the following table

String	a	B	c	a	b	c	a	b	c	a	b	c
p(P[1,…,i])												

(b) [14 marks] Describe your algorithm to compute p in general words and analysis its complexity