

Student ID: \_\_\_\_\_

Student Name: \_\_\_\_\_

**Part I. Filling-blank question [30 marks]**

1. What is the time complexity of deleting a node **p** from a doubly linked list? (**p** is neither the head nor the tail of the list)  $O(1)$  [3 marks]
2. The pop-out sequence of a stack is 1 3 2 4 5. Which of the following push-in sequence is/are impossible? CD [3 marks]  
A. 1 3 2 4 5    B. 1 2 3 4 5    C. 5 4 3 2 1    D. 5 3 2 4 1
3. (True or False) Suppose we implement a ring queue by an array **A** with size **m**, where **rear** and **front** are the index of the rear, and front of queue, respectively. Given the value pair of rear and queue. If **rear+1==front**, the queue is full. TRUE [3 marks]
4. What is the time complexity of the following method A. [3 marks]

```
method(n){  
    x=2;  
    while ( x < n/2 )  
        x = 2*x;  
}
```

  
A.  $O(\log n)$     B.  $O(n)$     C.  $O(n \log n)$     D.  $O(n^2)$
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 insert integer **k** into **A[i]** ( $i < n$ ), we need to move  $n-i$  integers. [3 marks]
6. The postfix expression of  $(4 - 3 - 2) / (6 * 5 + 1)$  is  $43-2-65*1+ /$ . [3 marks]
7. The following array is to be sorted in ascending order: 84, 47, 25, 15, 21. The process of sorting the order of the data is:  
step0: 84, 47, 25, 15, 21  
step1: 15, 47, 25, 84, 21  
step2: 15, 21, 25, 84, 47  
step3: 15, 21, 25, 47, 84  
step4: 15, 21, 25, 47, 84  
step5: 15, 21, 25, 47, 84  
The sorting algorithm used is/are A. [3 marks]  
A. Selection Sort    B. Bubble sort    C. Quicksort    D. Insertion sort
8. The feature of stack is(are) B. [3 marks]  
A. FIFO    B. FILO    C. LILO    D. RIO
9. Suppose that we use an array **A** [**0**, ... , **m-1**] to store the elements of a circular queue. If the head and tail pointers of the queue are front and rear, respectively, then the number of elements in the current queue is  $(rear+m-front) \% m$ . (If front == rear, the queue is empty.) [3 marks]
10. The number of prefixes for string "CS203" is C. (Excluding the empty string and the string itself.) [3 marks]  
A. 2    B. 3    C. 4    D. 5

## Part II. Short answer question [20 marks]

1. We assume  $\text{next}[0]=0$ . For example, the next array for aabaa is  $\{0,1,0,1,2\}$  .

1) What is the next array for abaabaab? [4 marks]

0,0,1,1,2,3,4,5

2) Please write down a string whose next array is  $\{0,0,1,0,1,2,3,2,3,1,2\}$ , only characters  $\{a, b, c\}$  can be used. [4 marks]

abacababaab

2. Function A is an implementation of binary search algorithm to find the smallest index of an integer  $k$  in non-descending size- $n$  array **Arr**. If  $k$  does not exist in **Arr**, return -1. But there are bugs in the code, please find them and fix them. [4 marks]

```
[1]  int A(int Arr[], int k)
[2]  {
[3]      int min = 0, max = Arr.length, mid;
[4]      while(min < max){
[5]          mid = min+(max - min)/2;
[6]          if (Arr[mid] < k){
[7]              min = mid;      min=min+1
[8]          }else{
[9]              max = mid - 1;  max=max
[10]         }
[11]     }
[12]     if(Arr[max] == k){
[13]         return max;
[14]     }else{
[15]         return -1;
[16]     }
[17] }
```

3. We want to design a queue by using two stacks. Please implement the “enqueue” and “dequeue” functions. You can and only can use `pop()`, `push(x)`, `isEmpty()` in Stack. The time complexities of `pop()`, `push(x)`, `isEmpty()` are  $O(1)$ .

*Note: We only give full marks to those solutions with optimal time complexity.*

Finish the following codes: [8 marks]

```
Stack S1,S2;
enqueue ( k )
{
```

s1.push(k)

```
}
```

```
dequeue()
{
```

```
    if(s2.isEmpty()){
        while(!s1.isEmpty()){
            s2.push(s1.pop());
        }
    }
```

```
}
```

```
    return S2.pop();
```

## Part II. Algorithm Design [50 marks]

**Note:** For each question in this part, Please design a correct algorithm for the given problem:

**First,** describe your ideas in general words in detail [70%]

**Second,** analyze the time complexity of your algorithm step by step [30%].

We only give full marks to those solutions with optimal time complexity.

1. [12 marks] Given an array  $A = \{a_1, a_2, \dots, a_n\}$  with size  $n$ . We say  $\langle a_i, a_j \rangle$  is a reversed pair iff  $a_i > a_j$  &  $i < j$ .

Please design an algorithm to find the total number of reversed pairs in array  $A$ , and return the sum of all reversed pairs.

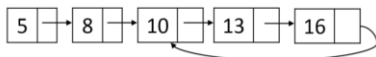
For example,  $A = \{3, 1, 2\}$ , the total number of reversed pair is 2 (i.e.,  $\langle 3, 1 \rangle$ ,  $\langle 3, 2 \rangle$ ), the sum of these 2 reversed pairs is  $(3 + 1) + (3 + 2) = 9$ .

```
mergesort
when merging
if a[l]>a[r]    count+=Lengthl-1    sum+=a[r]*(Lengthl-1)    r++
else if a[l]<a[r]    sum+=a[l]*r    l++
O(nlogn)
```

2. [12 marks] Design an algorithm to check whether a given linked list is a circular linked list or not.

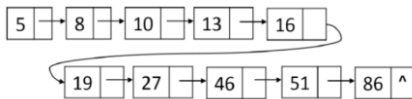
For example:

Linked list A is a circular linked list, return Yes.



A:

Linked list B is not a circular linked list, return No.



B:

```
fast=head;
slow=head;
while(true){
    slow=slow.next;
    if(fast.next!=null)
        fast=fast.next.next;
    else return No
    if(slow==null||fast==null) return No;
    if(slow==fast) return Yes;
}
O(n)
```

3. [12 marks] Given a String  $S$  and its length  $n$ , design an algorithm to calculate how many prefixes of  $S$  are not also a suffix of  $S$ . (Excluding the empty string and  $S$  itself.)

```
Get the next array of S.
such as abaaba's next array is 001123
k=next[n]
while(k!=0){
    count++;
    k=next[k];
}
count is how many prefixes of S is a suffix of S.
the answer is n-1-count
```

$O(n)$

4. [14 marks] Farmer John has built a new long barn, with  $N$  stalls. The stalls are located along a straight line at ascending positions  $x_1, \dots, x_N$ . The distance between  $x_i$  and  $x_j$  ( $j > i$ ) is  $x_j - x_i$ . His  $C$  ( $2 \leq C \leq N$ ) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, Farmer John want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

$O(n \log XN)$

```
min=0    max=XN-x1;
while(min<max){
    mid=(min+max)/2
    if(check(mid)==true) min=mid;
    else max=mid-1;
}
check(mid){
    i=0;j=0;count=0;
    while(true){
        while(xj-xi<=mid){
            j++;
            if(j>n){
                if(count>=c) return true
                else return false;
            }
        }
        count++;
        i=j-1;
    }
}
```