CS203 Data Structure and Algorithm Analysis Mid-term

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? C D [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. False [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;
}
```

$2^k = n/2 \Rightarrow k = \log_2 \frac{n}{2}$

   A. O(logn) B. O(n) C. O(nlogn) 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-k+1 integers. [3 marks]

6. The postfix expression of (4 - 3 - 2) / (6 * 5 + 1) is 2 43 - - 1 6 5 * + / .[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-front+m)%m .(If front == rear, the queue is empty.) [3 marks]

10. The number of true prefixes for string "CS203" is C .[3 marks]

    A. 2 B. 3 C. 4 D. 5

Part II. Short answer question [20 marks]

1. The next array for aabaa is {0,1,0,1,2} .

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

   [0,0,1,1,2,1,1,2]

   2) Please write down a string whose next array is {0,0,1,0,1,2,3,2,3,1,2}. [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]

```
int A(int Arr[], int k)       Arr.length-1
{                                 ^
    int min = 0, max = Arr.length, mid;
    while(min < max){ <=
        mid = min+(max - min)/2;
        if (Arr[mid] < k){
            min = mid;
        }else{                → mid+1
            max = mid - 1;
        }
    }              → max-1
    if(Arr[max] == k){
        return max; → max-1
    }else{
        return -1;
    }
}
```

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(S1.isEmpty()&&S2.isEmpty()) return null;
 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:

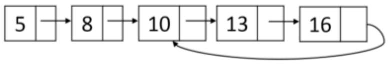
First, describe your ideas in general words in detail [70%]

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, … ,a_n} with size n. And reversed pair is <a_i,a_j> satisfied a_i>a_j && i<j. If there are totally k reversed pairs in A. Please design an algorithm to calculate k, and $\sum_{p=1}^{k} a_{i_p} + a_{j_p}$

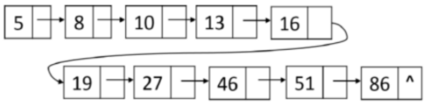
2. [12 marks] Design an algorithm to check if a linked list is a circular linked list.
   For example:
   Linked list A is a circular linked list, return Yes.



A:

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

快慢指针

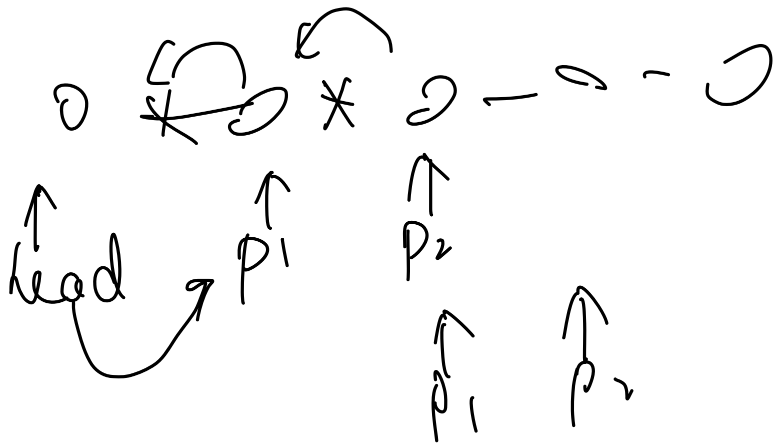


B:

```
( node head)

node slow = head;
node fast= head;   boolean che = false;
while( slow.next != null && fast.next.next !null){
        slow = slow.next;
        fast = fast.next.next;
        if (slow == fast){che = true ; break;}
} if (che) print("Yes");
      else pr ("No");
```

3. [12 marks] Given a String **S** and its length **n**, design an algorithm to calculate how many true prefixes of **S** are not also a true suffix of **S**.



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, ..., x_N. The distance between x_i and x_j (j > i) is x_j-x_i .
His C (2 <= C <= 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, FJ 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?

二分. 先假定结果，再断定

先给 l=0, r=INT_MAX, mid=l+(r-l)/2

对每个mid, 从X[0]开始,

若for (i: 0~C) inter=X[i]+mid 中 inter都小于等于X[ ]中的至少某一元素，则该mid可行，记录. (遍历判断)

若至少有一个 inter不小于等于X[ ]中某个元素，则该mid无效.

若最终mid > X[N-1], max=mid+1;

若最终mid < X[N-1], min=mid;

```
node  rev( node p1 , node p2, node head)
     p1 = head.next;
     p2 = p1.next;
     p1.next = head;
```