p M n ··· p m n    p N n ··· p L n

# 《数据结构与算法》第一次作业

## 一、选择题

C

1. 现有两个带头结点的双向循环链表，其头结点分别为 M 和 N，现将头结点为 N 的链表接到头结点为 M 的链表尾部，L 为 N 链表的最后一个结点，则相应的指针操作为（ C ）

A. L->next=M; M->prior=L; M->next->next=N; N->prior=M->next;

B. L->next=M; M->next->next=N; N->prior=M->next; M->prior=L;

C. L->next=M; M->prior->next=N; N->prior=M->prior; M->prior=L;

D. L->next=M; M->prior=L; M->prior->next=N; N->prior=M->prior;

C

2. 设栈 S 和队列 Q 的初始状态均为空，元素 abcdefg 依次进入栈 S。若每个元素出栈后立即进入队列 Q，且 7 个元素出队的顺序是 bdcfeag，则栈 S 的容量至少是（ C ）。

A. 1　　B. 2　　C. 3　　D. 4

A

3. 假设 $a_1,a_2,\ldots,a_n$ 依次入栈（且 $a_1..a_n$ 是 1..n 的排列），出栈为 1,2,…,n。以下哪种说法正确？（ A ）

$a_i \cdots a_j \cdots a_k \rightarrow a_k \cdots a_j \cdots a_i$

A. 不存在 i<j<k，使得 $a_i<a_j<a_k$.　　B. 不存在 i<j<k，使得 $a_k<a_j<a_i$.

C. 不存在 i<j<k，使得 $a_k<a_i<a_j$.　　D. 不存在 i<j<k，使得 $a_j<a_i<a_k$.

C

0 1 2 3 4 5

4. 若用一个大小为 6 的数组来实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，当从队列中删除一个元素，再加入两个元素后，rear，front 的值分别为多少？（ C ）

A.1 和 5

B.2 和 4

C.4 和 2

D.5 和 1

| front | rear |
|---|---|
| 3 | 0 |
| 4 | 0 |
| 4 | 1 |
| 4 | 2 |

D

BCDEFEF

5. 设串 s1= “ABCDEFG”，s2= “PQRST” 函数 strconcat（s，t）返回 s 和 t 串的连接串，strsub（s，i，j）返回串 s 中从第 i 个字符开始的、由连续 j 个字符组成的子串。strlength(s)返回串 s 的长度。则 strconcat（strsub（s1，2，strlength（s2）），strsub（s1，strlength（s2），2））的结果串是（ D ）。

BCDEF 5　　EF 5

A. BCDEF　　B. BCDEFG　　C. BCPQRST　　D. BCDEFEF

B

6. 如果主串和模式串的长度分别为 n 和 m，预处理模式串的 failure function 需要 B 时间。

A. O(n)　　B. O(m)　　C. O(n+m)　　D. O(nm)

## 二、简答题

1. 请用递推计算串“a b a a b a a b a b a”的 failure function π。

0 0 1 1 2 3 4 5 6 2 3

2. 已知线性表中的元素以值递增有序排列，并以**单链表**做存储结构。试写一高效的算法，

a< <b 且递增，找到 ≤a的地址，再找到<b部分

释放中间结点，连接，时间复杂度 O(n)

删除表中所有值**大于 a 且小于 b** 的元素（若存在），同时释放被删结点的空间，并分析时间复杂度（注意：a 和 b 是给定的两个参变量，它们的值可以和表中的元素相同，也可以不同）

3(Bonus 问题). 已知 Ackermann 函数定义如下:

$$\mathrm{Ack}(m,n)=\begin{cases} n+1 & 当m=0时 \\ \mathrm{Ack}(m-1,1) & 当m\neq 0,n=0时 \\ \mathrm{Ack}(m-1,\mathrm{Ack}(m,n-1)) & 当m\neq 0,n\neq 0时 \end{cases}$$

1) 写出计算 Ack(m,n) 的递归算法，并根据此算法给出出 Ack(2,1) 的计算过程。

2) 写出计算 Ack(m,n) 的非递归算法。

1) 见代码 Ack(2,1) → Ack(1, Ack(2,0)) → Ack(1,3) → Ack(0, Ack(1,2))

Ack(2,0) = 3: Ack(1,1) 3; Ack(0, Ack(1,0)) → Ack(0,2) → 3; Ack(1,0) → Ack(0,1) → 2

Ack(1,2) = 4: Ack(0, Ack(1,1)), Ack(1,1) = 3; Ack(0,3) 4

Ack(0,4) → 5

2) 见代码

提交时间 10 月 27 日 24 点前

提交地址：chenhd33@mail2.sysu.edu.cn