

## 2024 年 PKU 数算 A 期中考试

1. (2 分) 抽象数据类型的组成部分不包括

- A. 一组操作      B. 数据的对象      C. 数据的储存      D. 数据的关系

2. (2 分) 如下代码含义为, 找出  $\leq N$  的所有整数对  $(i, j)$ , 使得  $i * j \leq N$ , 它的时间复杂度是

```
for(int i=1;i<=N;i++){
    int max_j=N/i;
    int j=1;
    while(j<=max_j){
        printf("%d,%d",i,j);
        j+=1;
    }
}
```

- A.  $O(N \log(N))$       B.  $O(N^{1.5})$       C.  $O(N^2)$       D.  $O(\log(N)\sqrt{N})$

3. (2 分) 双向链表中, 若想交换结点  $p$  和它的后继, 且已知  $p$  的前驱和后继都存在,  $p$  的后继不是该链表的尾元素, 则以下正确的是

- A.  $p \rightarrow \text{prev} \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{prev} = p \rightarrow \text{prev}; p \rightarrow \text{next} \rightarrow \text{next} \rightarrow \text{prev} = p; p \rightarrow \text{next} \rightarrow \text{next} = p; p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{prev} = p \rightarrow \text{next};$
- B.  $p \rightarrow \text{next} \rightarrow \text{next} \rightarrow \text{prev} = p; p \rightarrow \text{prev} \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{next} = p; p \rightarrow \text{next} \rightarrow \text{prev} = p \rightarrow \text{prev}; p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{prev} = p \rightarrow \text{next};$
- C.  $p \rightarrow \text{next} \rightarrow \text{next} \rightarrow \text{prev} = p; p \rightarrow \text{next} \rightarrow \text{prev} = p \rightarrow \text{prev}; p \rightarrow \text{prev} = p \rightarrow \text{next}; p \rightarrow \text{prev} \rightarrow \text{prev} \rightarrow \text{next} = p \rightarrow \text{prev}; p \rightarrow \text{next} = p \rightarrow \text{prev} \rightarrow \text{next}; p \rightarrow \text{prev} \rightarrow \text{next} = p;$

4. (2 分) 下列正确的是

- A. 一个线性表长度已知, 且将来没有增删, 则从占用内存的角度考虑, 链表是比顺序表更好的选择
- B. 若一个长度为  $n$  的顺序表已经有序, 则插入一个新元素, 使其仍然保持顺序的时间复杂度是  $O(\log(n))$
- C. 在已经定位到待删除元素之后, 顺序表的删除操作时间复杂度是  $O(1)$
- D. 对于读操作密集, 很少发生增删的场景, 应选择顺序表

5. (3 分) 给定值为  $1 \sim 5$  的 5 个结点, 将他们组成一棵二叉搜索树, 并要求对于树上的每个节点, 其左子树的节点数不少于右子树的节点数。共可以组成\_\_\_\_\_棵二叉搜索树。

6. (6分) 用自底向上 SiftDown 方式构建最小堆，将序列 [23, 12, 17, 5, 28, 30, 15, 8] 构建成最小堆，得到的堆为\_\_\_\_\_。若现在向该堆中插入元素 4，经过 SiftUp 操作后，得到的堆为\_\_\_\_\_。

7. (3分) (多选) 设一棵树的嵌套表示法为 (A(B(D)(E(I)(J))(F))(C(G)(H))), 则该树没有几度结点

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

8. 假设以 S 和 X 分别表示入栈和出栈操作，则初态和终态均为空栈的入栈和出栈的操作序列可以表示为仅由 S 和 X 组成的序列。称可以操作的序列为合法序列（例如，SXSX 合法，SXXS 非法）。

(1) (2分) 试给出区分给定序列为合法或非法的充要条件（无需证明）。

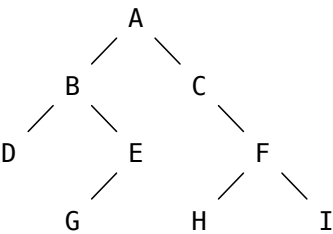
(2) (4分) 对元素不重复的输入序列，两个不同的合法序列可能得到相同的输出序列吗？请给出理由。

9. (1) (2分) 在实现 strcpy(char\* destination, char\* source)的过程中，假如输入的两个地址满足  $source < destination < source + strlen(source)$ ，此时该如何处理，为何？

(2) (4分) 计算字符串 abababa 的优化前后的 next 数组。

	a	b	a	b	a	b	a
前							
后							

10. (1) (3分) 给定二叉树，写出其前，中，后序遍历结果。



(2) (4分) 已知某二叉树前序序列为 ABDEGHCF，中序序列为 DBGEHACF，请写出该二叉树的后序序列。

11. 某公司需要对文件中的字符进行压缩编码。已知文件包含字符 a, b, c, d, e, f, g，且它们的出现频率分别为 6%, 8%, 9%, 10%, 12%, 25%, 30%。请回答以下问题：

(1) (3 分) 请给出该文件的哈夫曼树。(左节点权重小于等于右节点)

(2) (2 分) 使用哈夫曼编码, 该文件平均编码长度是多少?

(3) (3 分) 若文件中字符 g 出现的频率下降, 其它字符出现的比例不变, 哈夫曼编码方式不变, 当字符 g 的频率下降到多少时, 固定长度编码反而比哈夫曼编码更节省空间?

12. (3 分) 对下列 9 个等价对进行合并, 给出所得等价类树的图示。在初始情况下, 集合中每个元素分别在独立的等价类中。使用重量权衡合并规则, 合并时子树节点少的并入节点数多的那棵 (多的那个作为新树根, 少的那个作为新根的直接子节点); 若两棵树规模同样大, 则把根值较大的并入根值较小 (新树根取值小的)。同时, 采用路径压缩优化。

(0, 2), (1, 2), (3, 4), (4, 5), (2, 5), (6, 7), (8, 7), (2, 8), (9, 7)

13. 算法填空 (每空 2 分, 共 10 分)

以下是使用两个队列实现栈 (MyStack) 类的主要代码, 假设存放的元素均为 int 类型。其中 queue 可以使用的操作有:

push(x): 将元素 x 入队  
pop(): 队首元素出队  
front(): 返回队首元素的值  
empty(): 判断队列是否为空

```
class MyStack{
    queue<int> queue1, queue2;
    .....
    void push(int x){
        ____;
        while(____){
            ____;
            queue1.pop();
        }
        swap(queue1, queue2);
    }
    int pop(){
        int r=queue1.front();
        queue1.pop();
        return r;
    }
    int top(){
        return ____;
    }
    bool empty(){
        return ____;
    }
};
```

14. (10 分) 给定一个非空字符串, 判断它是否可以由它的一个子串重复多次构成。

样例 1: 输入 abab, 输出 True。解释: 由 ab 重复 2 次构成

样例 2: 输入 aba, 输出 False

样例 3: 输入 abcabcbcabcb, 输出 True。解释: 由 abc 重复 4 次构成

15. (10 分) 假定用两个一维数组 L 和 R 作为有 N 个结点的二叉树的存储结构。L[i] 和 R[i] 分别指示结点 i 的左儿子和右儿子的编号; 等于 0 表示为空。(结点编号从 1 开始

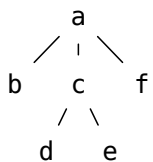
L[0]=R[0]=0)

(1) (6 分) 设计一个由 L 和 R 建立一个一维数组 T[n] 的算法, 使 T[n] 存放结点 i 的父亲 (T[0]=0)。

(2) (4 分) 根据 L, R, T 设计一个判别节点 U 是否为结点 V 后代的算法。

16. (10 分) 一棵树的镜面映射指的是对于树中的每个节点, 都将其子节点反序。我们在输入输出一棵树的时候, 常常会把树转换为对应的二叉树, 而且对该二叉树中只有单个子节点的分支节点补充一个虚子节点 \$, 形成“伪满二叉树”。对这棵二叉树进行前序遍历, 若是内部节点标记为 0, 若是叶节点标记为 1, 而且虚结点也输出。现在将“伪满二叉树”的前序遍历结果作为输入, 要求设计算法输出这棵树的镜面映射的宽度优先遍历序列。

对图中例子, 输入 a0 b0 \$1 c0 d0 \$1 e1 f1 \$1, 输出 a f c b e d



17. (10 分) 请证明: 对于非空满 k 叉树, 若其内部节点数目为 n, 那么其叶节点数目为  $n(k-1)+1$ 。