姓名：______ | 2 | 0 | | | | | | | | | ______ 班

题号	1	2	3	4	5	6	7	8	9	∑
得分										
题分										

1. 判断（请使用O、X）

☐ ……。
☐ ……。
☐ ……。
☐ ……。
☐ ……。
☐ ……。
☐ ……。
☐ ……。

2. 填空

A) ……（…）。
B) ……（…）。
C) ……（…）。
D) ……（…）。
E) ……（…）。
F) ……（…）。

3. RPN：试将中缀表达式“(0!+1)*2^(3!+4)-(5!/6+(7-(8-9)))”转换为逆波兰表达式

4. KMP：试分别给出如下模式串对应的next[]表和改进后的next[]表

j	0	1	2	3	4	5	6	7	8	9	10	11	12
P[j]	C	B	C	B	A	C	D	C	B	F	B	E	A
next[j]													
improvedNext[j]													

5. Recurrence：试给出如下T(n)的解析解，并说明理由

A) $T(n) = \begin{cases} \mathcal{O}(1) & (n \leq 1) \\ 2020 \cdot T(n^{1/2020}) + \mathcal{O}(\log n) & (n \geq 2) \end{cases}$

B) $T(n) = \begin{cases} \mathcal{O}(1) & (n \leq 1) \\ 2^{47} \cdot T(n/2^{2021}) + \mathcal{O}(\sqrt[43]{n}) & (n \geq 2) \end{cases}$

6. 简答（至多两行文字作答）

A) ......？

B) “在线”算法与“脱机”算法有何区别？

C) ......？

D) 如何从B-树中删除一个属于内部节点而非叶节点的关键码？

E) ......？

F) 按照Tarjan的定义，由n个节点组成的伸展树所具有的最大势能是多少？这类树是什么姿势？

G) ......？

7. Bidirectional Quadratic Probing (Two-Square Theorem of Fermat)

若采用“双向平方试探”策略的散列表长度取作$M = 2021$，则关键码0及其同义词所对应的试探链可记作：

$$a_0 = b_0 = 0,\ a_1 = 1,\ b_1 = 2020,\ a_2 = 4,\ b_2 = 2017,\ a_3 = 9,\ ,b_3 = 2012,\ a_4 = 16,\ ,b_4 = 2005,\ \ldots$$

试问：$\{\ a_i \mid i = 0, 1, 2, 3, \ldots\ \} \cap \{\ b_j \mid j = 0, 1, 2, 3, \ldots\ \}$除0以外，还包含哪些试探位置？为什么？

8. Reconstruction

```
// 由先序、中序遍历序列，重构一棵由n个节点组成的二叉树
BinNodePosi<T> reconstruct( T pre[], T in[], int n ) {
   if ( n < 1 ) return NULL;
   k = search( pre[0], in, n ); //locate pre[0] in in[] by SEQUENTIAL search
   return   new BinNode(
               pre[a], //key
               reconstruct( _____________, _____________, _____________ ) //left subtree
               reconstruct( _____________, _____________, _____________ ) //right subtree
            );
}
```

A）试补全递归调用处缺失的参数；B）设每次search()的返回值随机分布于[0,n)，试估计算法的期望时间复杂度。

9. Bitmap & QuadTree

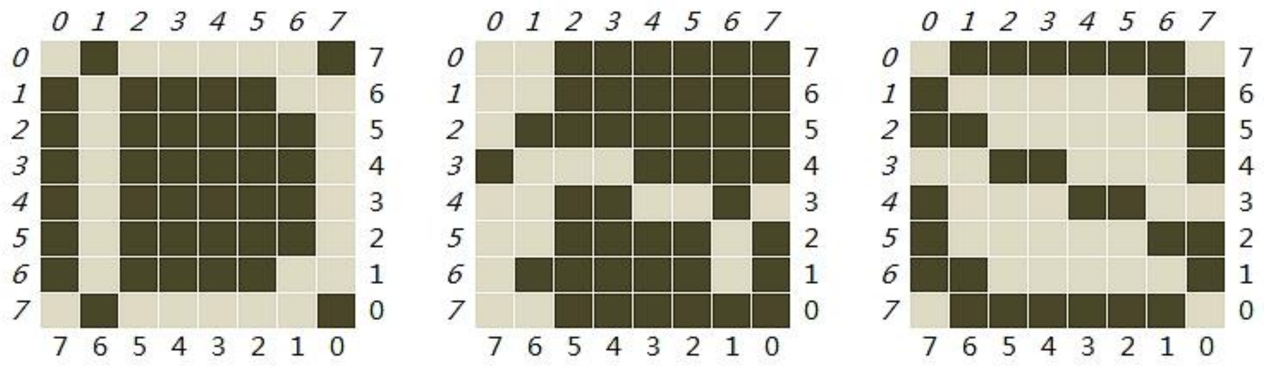
各像素或黑或白的图像，也称作**位图**；正方形的位图，可用**四叉树**来表示和存储——其节点可定义如下：

```
struct QuadNode {
   bool black; //若是叶节点，则非黑即白；对于内部节点，无意义
   QuadNode* child[4]; //按象限排序
   QuadNode( bool b ) //按指定颜色创建叶节点
      { black = b; for ( int i = 0; i < 4; i++) child[i] = NULL; }
};
```

（子）位图与（子）四叉树的对应规则为：

- 如果**位图**中的所有像素同色，则对应于一个该颜色的**QuadNode**
- 否则，将该**位图**平均划分为4块**子位图**（依次对应于四个象限），分别对应于一棵**子四叉树**

A） 试画出如下中间那幅**位图**所对应的四叉树（内部节点用$\boxed{\times}$标识，黑、白叶节点分别用$\boxed{1}$、$\boxed{0}$标识）：



B）试完成如下算法（**关键之处**需注释说明；**伪代码**即可，不必拘泥于C++的语法细节；如有必要，可补充子程序）

```
QuadNode* XOR( QuadNode* A, QuadNode* B );
    //A、B为大小相同的两幅位图对应的四叉树，试计算出二者的叠合位图所对应的四叉树
    //所谓叠合，即两幅位图中对应的像素分别做一次异或运算（如上的中图，即为左图、右图的叠合）
```