姓名：

2	0								

题号	1	2	3	4	5	6	Σ
得分							
题分	12	48	16	8	6	10	100

---

0. 预备

关闭手机、计算器等电子设备； 确认总共4页，无缺页、错页； 在卷首注明你的姓名和学号

题中所指页码，均是对讲义打印版而言； 凡交待未尽之处，皆以讲义及示例代码为准

充分利用好草稿纸，保持卷面的清晰、整洁

1. 判断（涂黑你的选项） 2 ×6

[T] [F] 即便$f(n) = \mathcal{O}(g(n))$，也未必$2^{f(n)} = \mathcal{O}(2^{g(n)})$。

[T] [F] 不存在CBA式算法，能够经过少于2n-3次比较操作，即从n个整数中找出最大和次大者。

[T] [F] 存在CBA式算法，能够在$\mathcal{O}(n)$时间内从n个无序整数中找出最大的10%。

[T] [F] 起泡排序过程中，每经过一趟扫描交换，相邻的逆序对必然减少。

[T] [F] 即便借助二分查找确定每个元素的插入位置，向量的插入排序在最坏情况下仍需$\Omega(n^2)$时间。

[T] [F] 带权重的最优PFC编码树不仅未必唯一、拓扑结构未必相同，甚至树高也可能不等。

2. 选择（请列出代号；可能有多个选项） 4 ×12

1) 若每一递归实例本身仅需常数时间和空间，则（　　）函数的渐进时间复杂度等于渐进空间复杂度。
 A) 尾递归　B) 线性递归　C) 二分递归　D) 多分支递归

2) 使用binsearch算法版本C在有序向量{ 1, 3, 5, ..., 2013 }中查找，目标为独立均匀分布于[0, 2014]内的整数。若平均失败查找长度为F，则平均成功查找长度S应为（　　）。
 A) $\frac{1008F}{1007}+1$　B) $\frac{1008F}{1007}-1$　C) $\frac{1008(F-1)}{1007}+1$　D) $\frac{1008(F+1)}{1007}-1$

3) 设图灵机在初始状态下，只有读写头所对单元格为'0'，其余均为'#'；此后，连续地执行increase()算法2014次。在此期间，读写头累计移动的次数（就相对误差率而言）最接近于（　　）。
 A) 2,000　B) 4,000　C) 8,000　D) 16,000　E) 32,000

4) 字符串"123XY"中的字符经栈混洗之后，可得到（　　）个合法的C++变量名（比如"YX321"）。
 A) 28　B) 5　C) 6　D) 5　E) 以上皆非

5) evaluate()算法的优先级表中，有的空格项对应于表达式不合法或不合常识的情况，比如（　　　）。
A) pri['\0'][')']　　B) pri['!']['(']　　C) pri[')']['!']　　D) pri['(']['\0']

6) 实际上，evaluate()算法居然可以对非法表达式"(12)3+!4*+5"进行"求值"，其返回值为（　　　）。
A) 41　　B) 89　　C) 365　　D) 以上皆非

7) 若仅考查最好情况下的渐进时间复杂度，则(B)ubblesort（p163版）、(I)nsertionsort、(M)ergesort（p168+170版）、(S)electionsort的非降排列次序是（　　　）。
A) IBMS　　B) MIBS　　C) SMIB　　D) IMSB　　E) BIMS

8) （　　　）算法在最好情况与最坏情况下的渐进性能相同。
A) Bubblesort（p163版）　B) Insertionsort　C) Mergesort（p168+170版）　D) Selectionsort

9) 将有序列表L均分为长Θ(h)的k段，各段分别置乱。则L.insertionSort()至多只需（　　　）时间。
A) $\Theta(h^2 \cdot k^2)$　　B) $\Theta(h \cdot k^2)$　　C) $\Theta(h^2 \cdot k)$　　D) $\Theta(h \cdot k)$

10) 若将有根有序的多叉树T所对应的二叉树记作B(T)，
则T的（　　　）遍历序列与B(T)的（　　　）遍历序列完全相同。
A) 后序...后序　　B) 后序...中序　　C) 层次...先序　　D) 先序...先序　　E) 以上皆非

11) 在二叉树（　　　）遍历序列中，祖先节点一定位于其后代节点之前。
A) 先序　　B) 中序　　C) 后序　　D) 层次　　E) 以上皆非

12) 在Huffman编码算法中，若每次（超）字符合并时均保证左兄弟不小于右兄弟，
则在所生成编码树的层次遍历序列中，（　　　）必然按其频率的非升次序排列。
A) （仅）叶节点　　B) （仅）内部节点　　C) 所有节点　　D) 以上皆非

3. 填空（无需给出计算过程；如认为有歧义，可做扼要说明）　　4 x4

1) 表达式"(0 ! + 1) * 2 ^ (3 ! + 4) - 5 / (6 ! / 7 !) - 8 + 9"所对应的RPN为：

2) 对由2014个节点构成的完全二叉树做层次遍历，辅助队列的容量至少应为（　　　）；
在整个遍历过程中，辅助队列的规模共在（　　　）步迭代中处于这一规模。

3) 据大道消息，某官员因家中被起获2亿余元，被罚将这些钞票（均为百元面额真币）按编号手工排序。
若他只懂基本的起泡排序算法，那么即便每秒可完成一次比较和交换，亦大致耗时（　　　）世纪。

4) 设在List::selectionSort()算法中，将：

```
insertB( tail, remove( selectMax( head->succ, n ) ) );
```

替换为：

```
swap( tail->pred->data, selectMax( head->succ, n )->data );
```

若输入列表为{ 1962, 1963, ..., 2014; 1, 2, 3, ..., 1960, 1961 }，
则swap()语句无实质效果（原地交换）的情况共计出现（　　　）次。

4. 计算（保留推导过程，包括图、表，这些是更重要的评分依据） 4 ×2

设整数e独立且均匀地取自[0, 25)，现通过调用fibSearch(A, e, 0, 7)，对如下整型向量A[]做查找：

k	*0*	*1*	*2*	*3*	*4*	*5*	*6*
A[k]	1	3	5	7	9	17	19

试分别计算其在失败情况下的平均查找长度，以及总体的平均查找长度。

5. 证明（请同时给出示意图） 6

在由n个节点构成的二叉树中，任意节点v_i和v_j之间的距离取作二者之间那条唯一通路的长度，记作$\|v_iv_j\|$。

试证明：若二叉树的先序遍历序列为$\{ v_0, v_1, v_2, \cdots, v_{n-1} \}$，则有：

$$\sum_{k=0}^{n-1} \|v_k v_{(k+1) \bmod n}\| = \underbrace{\|v_0v_1\| + \|v_1v_2\| + \cdots + \|v_{n-2}v_{n-1}\| + \|v_{n-1}v_0\|}_{n} = 2 \cdot (n-1)$$

6. 算法 2 + 3 + 5

以下代码中的 int parent[0, n) ，是采用父节点表示法存储的任意一棵有根（但未必有序）的多叉树。

```
int f( int parent[] , int n ) { //-1 < n
    int *p= new int[n];
    for (int i = 0; i < n; ++i) p[i] = parent[i];
    int h = -1 ;
    parent = p;

    for ( int i = 0 ; i < n ; i ++ )


        h = __max ( h , g( parent , i ) ) ;

    --h;
    return h ;
}
int g( int parent[] , int i ) {
    if(parent[i] < 0) return -parent[i];

    if ( -1 == i ) return -1 ;
    parent[i] = -(1 + g(parent, parent[i]));

    return 1 + g( parent , parent[ i ] ) ;
}
```

A) 以上算法f()和g()分别是何功能？

B) 在最坏情况下，算法f()的渐进时间复杂度是多少？最坏情况何时出现？

C) 在不做任何删除的前提下，试通过增加尽可能少的代码，使f()的运行时间降至$O(n)$，空间不超过$O(n)$。简要说明你的改进策略与思路，然后直接在原代码基础上完成修改，并为关键环节增加注释。