2015-2016 学年第 1 学期考试试题　（A）卷

参考答案和评分标准

一. 单选题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B	D	D	B	A	C	B	D	D	A	A	C	C	B	C

二. 填空题（每题 1 分，共 10 分）

1. 10^3　nlogn　n^2　2^n
2. 递归结束条件 / 递归出口
3. O（$n^{\log_{\frac{3}{2}}\frac{8}{3}}$）
4. 20
5. 23
6. 约束函数
7. 重排
8. 队列式(FIFO)
9. 13
10. input(a,n-1)

三. 算法应用题（每题 8 分，共 40 分）

1. 覆盖方案：其中阴影方格为特殊方格（标注出 L 型骨牌序号，从 0 号开始）。

2	2	4	4	12	12	14	14
2	1	1	4	12	11	11	14
3	1	5	5	13	13	11	15
3	3	5	0	0	13	15	15
7	7	9	9	0	17	19	19
7	6	9	■	17	17	16	19
8	6	6	10	18	16	16	20
8	8	10	10	18	18	20	20

2.

最优调度方案为（6 分）

2	7	5	4	8	1	6	3

所需的最少时间为：73　（2 分）

3. ① 首先，将这 10 位客户的申请按照结束时间 f(i)递增排序（3 分）

t[i]	3	2	1	6	5	4	10	9	8	7
s(t[i])	1	3	0	5	3	5	6	8	8	11
f(t[i])	4	5	6	7	8	9	10	11	12	13

② 选择申请 3(1,4)　（1 分）

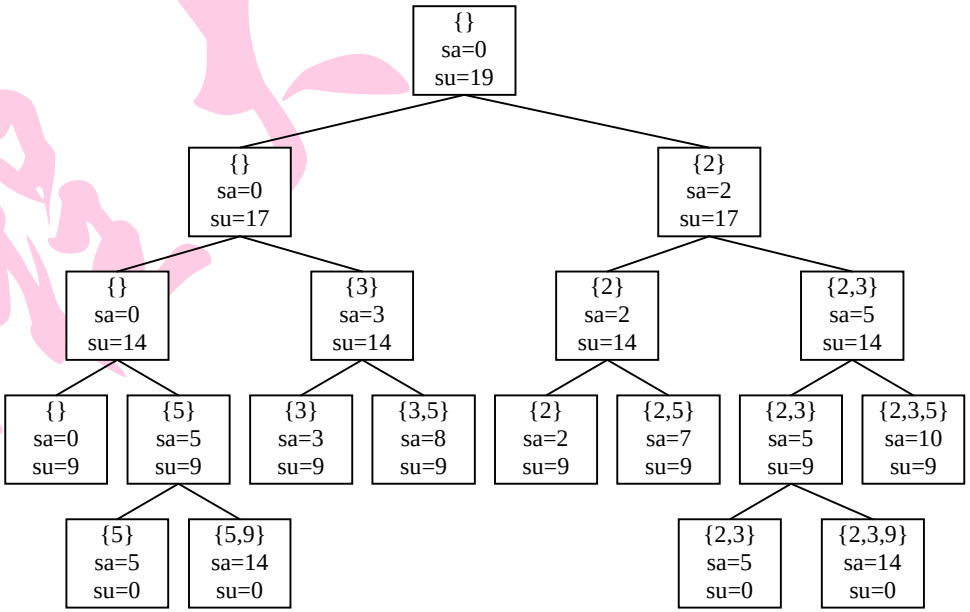
③ 依次检查后续客户申请，只要与已选择的申请相容不冲突，则选择该申请。

直到所有申请检查完毕。申请 6(5,7)、申请 9(8,11)、申请 7(11,13)（3 分）

④ 最后，可以满足：3(1,4)、6(5,7)、9(8,11)、7(11,13)共 4 位客户的申请，最多可以安排 4 位客户的申请。（1 分）

4. 状态空间树：（5 分）

设 sa 为当前状态的和，su 为剩余元素的和



回溯求解的过程：（3 分）

沿状态空间树深度优先搜索，左分枝表示放弃，右分枝表示选入。

在结点({}/sa=0/su=9)处，因 sa+su<M，回溯；

在结点({5}/sa=5/su=0)处，因 sa+su<M，回溯；

在结点({5,9}/sa=14/su=0)处，得解；

在结点({3}/sa=3/su=9)处，因 sa+su<M，回溯；

在结点({3,5}/sa=8/su=0)处，因 sa+A[3]>M，回溯；

在结点({2}/sa=2/su=9)处，因 sa+su<M，回溯；

在结点({2,5}/sa=7/su=9)处，因 sa+A[3]>M，回溯；

在结点({2,3}/sa=5/su=0)处，因 sa+su<M，回溯；

在结点({2,3,9}/sa=14/su=0)处，得解；
在结点({2,3,5}/sa=10/su=9)处，因 sa+A[3]>M，回溯。
最后得解：{5,9}, {2,3,9}

5. （1）贪心策略：最短程序优先。将程序从小到大排序，依次选取尽可能多的程序，但总长度不超过磁盘容量，则可求得最多可以存储的程序个数 m。（4 分）
（2）采用回溯法，从 n 个程序中选取总长度最大的 m 个，算法同装载问题。（4 分）

四．算法设计题（使用 C 或 C++或 Java 语言实现）（每题 10 分，共 20 分）

1. （1）算法设计：

```
void Knapsack(int n,float M,float v[],float w[],float x[]){
    int *t = new int [n+1];                  //2 分
    Sort(n,v,w,t);                           //1 分
    for ( int i=1;i<=n;i++) x[i]=0;          //1 分
    float c=M;
    for (i=1;i<=n;i++) {
        if (w[t[i]]>c) break;
        x[t[i]]=1;
        c-=w[t[i]];
    }                                        //2 分
    if (i<=n) x[t[i]]=c/w[t[i]];             //2 分
}
```

（2）算法分析：算法的时间复杂度为 O（n*logn）
因为排序算法的时间复杂度为线性对数阶。 //2 分
注：未定义下标数组，扣 3 分

2. 求数塔一条路径，使路径上的数值和最大

```
void main( ){
    int a[50][50][3],i,j,n;
    printf("please input the number of rows:");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++) {
            scanf("%d",&a[i][j][1]);
            a[i][j][2]=a[i][j][1];
            a[i][j][3]=0;
        }
    }                                        //2 分
    for(i=n-1;i>=1;i--)    {
        for(j=1;j<=i;j++){
            if(a[i+1][j][2]>a[i+1][j+1][2]) {
                a[i][j][2]=a[i][j][2]+a[i+1][j][2];
                a[i][j][3]=0;
            }
            else {
                a[i][j][2]=a[i][j][2]+a[i+1][j+1][2];
                a[i][j][3]=1;
            }
        }
    }                                        //5 分
    printf("max=");
    j=1;
    for(i=1;i<=n-1;i++)    {
        printf("%d->",a[i][j][1]);
        j=j+a[i][j][3];
    }
    printf("%d",a[n][j][1]);                 //3 分
}
```