算法设计与分析期中考试试卷

答题要求：解答算法设计题目时，请先用一段话描述算法思想。若用动态规划算法，请写出递推方程、边界条件、标记函数等设计要素；贪心法需给出证明；回溯法需给出解向量、搜索树、约束条件、优化算法等；各种算法需分析时间复杂度。阅卷时会根据算法的正确性和效率评分。

一、（10 分）求解递推方程，每题 5 分

1. $T(n) = T(n-1) + \log n$, $T(1) = 1$
2. $T(n) = 4\,T(n/2) + n^2$，$T(1) = 1$

答案：

1. 迭代法， 推出 $T(n)= \log n! = \Theta(n \log n)$

2. 根据主定理第二种情况， a=4, b=2, $n^{(\log_b a)}=n^2$, $T(n) = \Theta(n^2 \log n)$

二、（10 分）下面是算法 ALG 的伪码，输入 A 是 n 个互不相等的浮点数的数组。请说明该算法输出的 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 分别是具有什么性质？以浮点数比较计算为基本操作，该算法的时间复杂度（渐进复杂度）是多少？该算法精确的浮点数比较次数是多少？

```
void ALG(double A[], int n) {
  double x, y;
  x = y = a[n];
  for (int i = 1; i < n; i += 2) {
    if (a[i] > a[i+1]) {
      if (x < a[i])
        x = a[i];
      if ( y > a[i+1])
        y = a[i+1];
    }
    else {
      if (x < a[i+1])
        x = a[i+1];
      if (y > a[i])
        y = a[i];
    }
  }
  printf("%lf %lf\n", x, y);
}
```

x 和 y 分别是 A 数组中的最大值和最小值

$$\boldsymbol{\Theta(n)}$$

$$\boldsymbol{3\left\lfloor\frac{n}{2}\right\rfloor}$$

三、（10 分）

某同学大四最后一学期选课，根据培养方案，他还需要选 5 学分的专业限选课，3 学分的通选课，2 学分的公共基础课才能毕业。现在他感兴趣的课表如下（假定课表上的上课时间都不冲突）：

课号	类别	学分	学时
1	通选	3	3
2	专业限选	2	3
…			
n	公共基础课	3	4

同时，他想尽可能的减少上课时间，以便有更多的时间来写本科毕业论文。请问如何选择课程，来实现上述目标？写出模型即可，不用具体求解。

1. 请先定义变量和写出目标函数。（5 分）

   答案：Xi 代表是否选这门课，Ti 代表学时，Si 代表学分，Di 代表通选课学分，Zi 专业限选学分，Gi 代表公共基础课学分，如果不是这个类别则为 0

   $$\min z = \sum_{i=1}^{n} x_i T_i$$

   错一个变量扣 0.5 分，目标函数错扣 2 分

2. 请写出要满足的约束条件。（5 分）

   $$\sum_{i=1}^{n} x_i D_i \geq 5,$$
   $$\sum_{i=1}^{n} x_i Z_i \geq 3,$$
   $$\sum_{i=1}^{n} x_i G_i \geq 2,$$
   $$x_i = 0,1$$

   如果错一个条件，扣 1 分

四、（10 分）

任意给 n 个正整数$a_1, a_2, ..., a_n$，设计一个算法判断能否把这些整数恰好分成和相等的两个部分？即存在子集$T \subseteq I = \{1,2, ..., n\}$ 使得 $\sum_{i \in T} a_i = \sum_{i \in I-T} a_i$ 。

如果用了动态规划，满分 7 分；如果是回溯，满分 10 分。

回溯：
设计解向量$(x_1, x_2, ..., x_n)$分量取值为 0 或 1，搜索空间为子集树。
$x_i = 1$ 表示选择a_i到子集中。

约束条件$\sum_{0 \le i \le n} x_i a_i \le \frac{1}{2} \sum_{0 \le i \le n} a_i$

优化：
对称性优化
a_1必定在某一组当中，向量可以减少到$n-1$ 维
排序优化
把整数从大到小排序，得到新的序列$a_1, a_2, ..., a_n$

五、（20 分）马上要开校运会了，假设你是比赛的组织者，共有 n 个选手。每个参赛选手都有一个编号 1，…，n。这些选手之间形成若干小组，允许一名选手属于多个小组。每个小组的选手编号都是连续的，例如{2,3,4}，{8,9,10,11}。现在你要选取若干名选手作为联络人，请设计一个算法找到最少的联络人，使得每个小组都至少有一名联络人。

答案： 贪心法 （得 5 分）

算法设计（4 分）

假设每个小组对应的编号区间为 A_i=[a_i,b_i]，按照每个小组最大编号的选手从小到大进行排序，先把 b_1 设为联络人，然后检查其他组的 a_i 是否小于 b_1，如果小于，则满足；否则，将 b_i 加入，继续检查。记最终选出的联络人集合为 S。

算法复杂度（1 分）如果有 m 个小组，算法复杂度为 O(mlog m).

正确性证明（10 分）

命题 4.20 按照上述算法选出的 S 是与每个 A_i 相交的最小集合.

证 对步数 k 归纳.

$k=1$，存在最优解 S 含有 b_1. 假设 T 是最优解，T 含有 A_1 中的 x_i. 如果 x_i 不是 b_1，用 b_1 替换 x_i 得到 T'，由于 $x_i<b_1$，每个集合是连续整数，含有 x_i 的集合一定含有 b_1，于是 T' 也是最优解.

假设对于任意 k，算法 k 步选择都导致最优解，即存在最优解

$$T = T_k \cup B$$

其中 $T_k=\{b_1,b_{i2},\cdots,b_{ik}\}$ 是算法前 k 步选择的数的集合. 如果 T_k 与 $A_{k+1},\cdots,A_n$ 不交的集合为 $A_{j1},A_{j2},\cdots,A_{jl}$，那么 B 是 A_{j1}，A_{j2}，$\cdots$，A_{jl} 的最优解. 若不然，存在更好的解 B^*，那么 $T_k \cup B^*$ 将是比 T 更好的解，与归纳假设矛盾.

根据归纳基础，$A_{j1},A_{j2},\cdots,A_{jl}$ 有一个最优解 B' 含有算法第 $k+1$ 步选择的元素 $b_{i(k+1)}$，且 $|B'|=|B|$，于是

$$T' = T_k \cup B' = T_k \cup \{b_{i(k+1)},\cdots\} = \{b_1,b_{i2},\cdots,b_{ik},b_{i(k+1)},\cdots\}$$

且 $|T'|=|T|$，于是算法到第 $k+1$ 步的选择也将导致最优解. 根据归纳法命题得证.

六、(20 分)对玻璃瓶做强度测试，设地面高度为 0，从 0 向上有 n 个高度，记为 $1,2,3,\ldots,n$，其中任何两个高度之间的距离都相等。如果一个玻璃瓶从高度 i 落到地上没有摔破，但从高度 $i+1$ 落到地上摔破了，那么就将玻璃瓶的强度记为 i。

（1）假设每种玻璃瓶有足够多的相同的测试样本，设计算法使用最少的测试次数来完成测试。该算法的最坏情况下的时间复杂度是多少？

（2）假设每种玻璃瓶只有 2 个相同的测试样本，设计次数尽可能少的算法完成测试。你的算法最坏情况下的时间复杂度是多少？

（1） $O(\log n)$ 二分测试，每次在 n/2 处测试，根据测试结果选择在某半边递归的测试即可

（2） $O(\sqrt{n})$ $\sqrt{n}$分测试，一个瓶子依次测$i\sqrt{n}$高度$i = 1,2,\ldots,\sqrt{n}$确定硬度区间，第二个瓶子测这个区间内的硬度

七、（20 分）热爱极限运动的小明计划参加一项公益跑步活动，在未来的 $\boldsymbol{n}$ 天里按预先确定的顺序到访 $\boldsymbol{n}$ 个城市做公益。假设第 $\boldsymbol{i}$ 天从前一个城市出发到第 $\boldsymbol{i}$ 个城市的距离为 $\boldsymbol{x_i}$。如果小明体力好，可以完全跑完这段路程；但如果跑不下来全程，也可以坐收容车完成当天剩余的路程。连续每天跑步会消耗小明的体力，假设小明连续 $\boldsymbol{n}$ 天跑步，每天能跑的路程为 $\boldsymbol{s_i}$，且满足

$$\boldsymbol{s_1 > s_2 > \ldots > s_n > 0}$$

但如果小明某天不跑，完全坐收容车完成当天路程，那么第二天小明的体力就可恢复如初，接下来得日子里又可以跑完$\boldsymbol{s_1, s_2}, \ldots$的路程了。为了在整个活动中跑出最长的总距离，小明需要仔细计划一下，都应该在哪些天里不跑步完全坐收容车完成当天路程。请为小明设计一个算法解决该问题。

$$t(i,k) = \sum_{1 \le l \le k} \min\{s_l, x_{i+l-1}\}$$

表示小明第$i-1$休息后连续跑到k天，在这k天中所能跑的最大路程之和

$$v(i) = \max_{0 \le j < i}\{v(j-1) + t(j+1, i-j)\}$$

$$m(i) = j\text{使得}v(i)\text{最大}$$

表示小明前i天中，不修整（$j=0$）或最后一次在第j天修整时，所能跑的最大路程之和

$$v(n)$$

为最大值

算法复杂性

$$O(n^2)$$