装订线内 不要答题

北京大学信息科学技术学院考试试卷

考试科目： 算法设计与分析 **姓名：**__________ **学号：**__________

考试时间： 2018年6月24日 **大班教师:**__________**小班教师:**__________

题号	一	二	三	四	五	六	七	八	**总分**
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过15分钟不得入场。在考试开始30分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

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、搜索树等、约束条件；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。

得分

一、判断题（10 分，每题 1 分， 正确打✓，错误打✘）

1. 如果 P=NP，那么每一个 NPC 问题都可以在多项式时间内求解。(✓)
2. NP-hard 问题一定是 NPC 问题。(✘)
3. 求 n 个数中的次大数最少只需要比较 log(n)次。(✘)
4. 近似算法的近似比越高，算法所得到的近似解越接近最优解。(✘)
5. $\log(n!)=\Theta(n\log(n))$ (✓)
6. 0-1 背包问题可以用动态规划方法求解，因此应该不属 NPC 类问题。(✘)
7. 一般货郎问题存在 2-近似算法。(✘)
8. 随机选择算法 RandomSelect 是一个 Monte Carlo 算法。(✘)
9. 线性规划问题一定有唯一的最优解。(✘)
10. 如果 f 是最大流，那么不存在关于 f 的 s-t 增广链。(✓)

得分

二、单选题（每小题 2 分，共 10 分）

1. 假设某算法的计算时间表示为递推关系式

$$T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n},\ T(1) = 1$$

则算法的时间复杂度为（ C ）。

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

2. 给定含有 n 个不同的数的数组 L=<x_1, x_2, ..., x_n>。如果 L 中存在 x_i（1 < i < n）使得 $x_1 < x_2 < \ldots < x_{i-1} < x_i > x_{i+1} > \ldots > x_n$， 则称 L 是单峰的，并称 x_i 是 L 的“峰顶”。现在已知 L 是单峰的，请把 a-c 三行代码补全到算法中使得算法正确找到 L 的峰顶。

a. Search(k+1, n)

b. Search(1, k-1)

c. return L[k]

```
Search(1, n)
k←⌊n/2⌋
if L[k] > L[k-1] and L[k] > L[k+1]
then __________
else if L[k] > L[k-1] and L[k] < L[k+1]
then __________
else __________
```

正确的填空顺序是（ A ）。

A. c, a, b　　B. c, b, a　　C. a, b, c　　D. b, a, c

3. 同时查找 2n 个数中的最大值和最小值，最少比较次数为（ C ）。

A. 3(n-2)/2　　B. 4n-2　　C. 3n-2　　D. 2n-2

4. 以下最坏时间复杂度不是 $O(n^2)$ 的排序方法是（ B ）。

A. 插入排序　　B. 归并排序　　C. 冒泡排序　　D. 快速排序

5. 以下问题中属于 NPC 的是（ B ）。

A. 最长公共子序列　B. 双机调度　C. 最小生成树　D. 快速排序

得分

三、（10 分）

问题：矩阵 A 有 m 行 n 列，其中每个元素均为 0 或 1。是否存在矩阵 A 的若干行，使得以这些行组成的子矩阵中每一列恰好有一个 1？例如，在下图的 5 行 6 列的矩阵中，选择 2，4，5 行（阴影部分），可以满足要求。

1	1	1	1	0	1
1	1	1	0	0	0
1	0	1	1	0	1
0	0	0	0	1	1
0	0	0	1	0	0

请设计一个算法解决该问题并分析时间复杂度。

参考答案（一）：

使用回溯算法。解向量<x_1, x_2, …, x_k>，表示当前搜索前 k 行是否选中。搜索树有 2^m 个节点。每个节点使用 O(n)的时间来检查第 k+1 行能否选中，以及计算覆盖状态。能达到覆盖所有列的状态，输出 YES；否则遍历完毕，输出 NO。

时间复杂度：最坏情况下为 O(n*2^m)

参考答案（二）

使用动态规划。

cover[c][r]，0≤c<2^n，1≤r≤m。若 cover[c][r]=1，表示矩阵前 r 行，能覆盖编码为 c 的列的组合；否则若 cover[c][r]=0，表示不能覆盖。若存在 r 使得 cover[2^n-1][r]=1，问题实例的答案是 YES。

算法伪代码如下（标记函数略）。

```
// 边界条件：1-5 行
1:  for ( 0  ≤  c < 2^n ) do
2:      cover[c][1] = 0;
3:  end for
4:  x = encode(row [1]);   // 第 1 行编码（例子中是 0x3D）
5:  cover[0][1] = cover[x][1] = 1;
// 递推方程：6-14 行
6:  for ( 1  ≤  r < m ) do
7:      for ( 0  ≤  c < 2^n ) do
8:          if (cover[c][r] == 1) then
9:              cover[c][r+1] = 1;   // 前 r 行能覆盖，前 r+1 行也能覆盖
// 第 r+1 行编码（例子中第 2 至 5 行分别是 0x38、0x2D，0x03，0x04）
```

```
            x = encode(row[r+1]);
           if (x & c == 0) then cover[x|c][r+1] = 1; end if   // 每列最多 1 个 1
        end if
    end for
end for
return (cover[2^n-1][m]==1) ? YES : NO;
```

时间复杂度：循环 m*2^n 次，每次位运算 O(n)，总共 O(m*n*2^n)时间
空间复杂度：优化需 O(2^n)。未优化 O(m*2^n)。

得分

四、线性规划应用（10 分）

期末算法小班聚餐，餐费标准是人均 50 元。现在决定从“站点比萨”购买比萨和饮料，可供选择的有五种比萨和两种饮料，如下表所示：

	比萨					饮料	
名称	垃圾桶	夏威夷	加州小牛肉	绞肉机	站点特色	可乐	雪碧
价格（元）	220	165	195	200	190	15	15
分量	6 人份	4 人份	5 人份	6 人份	5 人份	3 人份	3 人份

假设一个小班共有 15 人（包括助教和老师在内），请问如何选择比萨和饮料来满足以下要求：（1）每个人都至少有一份比萨和一份饮料；（2）人均消费不超过 50 元；（3）任何 1 种比萨的数量不能超过总比萨数量的一半；（4）总花费最少。

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：定义变量x_i分别表示五种比萨和两种饮料选择的数量，c_i表示对应的价格，q_i表示对应的份数。

   目标函数是要最小化 $\sum_{i=1}^{7} x_i c_i$

   评分标准：x 定义正确得 2 分，c 正确得 1 分，目标函数正确得 2 分。

2. 请写出要满足的约束条件。（5 分）

$$\sum_{i=1}^{5} x_i q_i \geq 15,\quad \sum_{i=6}^{7} x_i q_i \geq 15$$
$$\sum_{i=1}^{7} x_i c_i \leq 15 * 50,$$
$$\sum_{i\neq j, i\leq 5} x_i > x_j,\quad 任意\ 1\leq j \leq 5$$

评分标准：前三个条件各 1 分，最后一个 2 分

装订线内不要答题

得分

五、网络流应用（20 分）

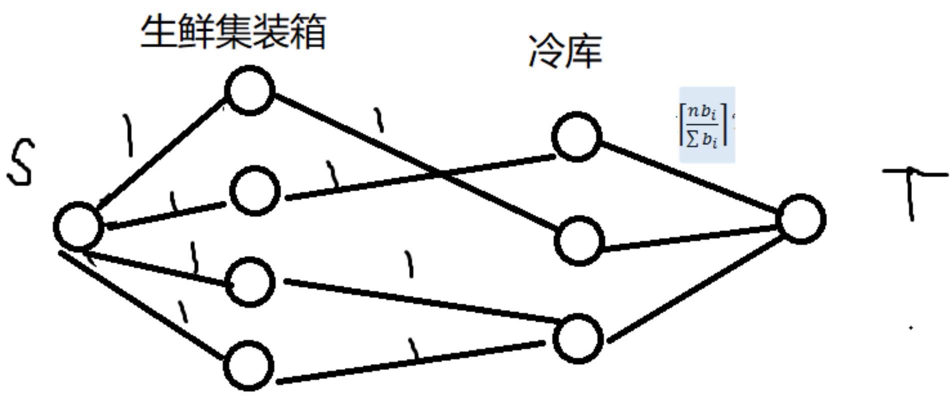
生鲜商品的物流问题，只有尽快把生鲜商品及时送到冷库才能最大限度的减少生鲜商品的变质。假设共有 k 个冷库，每个冷库 i 最多可以储存b_i个生鲜集装箱。现共有 n 个生鲜集装箱，每个集装箱 j 距离每个冷库 i 的距离以运送时间计算记为 t_{ij}。我们希望集装箱既能在 1 小时内送到冷库，又不能把一个冷库的储存空间用完导致无法储存后续出现的生鲜集装箱，因此需要尽可能地保持冷库空间占用率平均，不要过于集中。

1. 请设计一个最大流算法，判断是否可以将 n 个生鲜集装箱在 1 小时内送往 k 个冷库，并保证每个冷库 i 的存储空间占用率不超过$\left\lceil\frac{nb_i}{\sum b_i}\right\rceil$？画出流网络图，说明源点、汇点、中间节点、边、容量、流量分别代表的含义。（10 分）

答案：如下图所示，建立第一层节点为生鲜集装箱，第二层节点为冷库。对于每个集装箱所对应的顶点，如果该集装箱能够在 1 小时内被送达某个冷库，那么将该集装箱顶点连边到该冷库节点。源点到集装箱节点的边的容量为 1，集装箱节点到冷库节点的边的容量也为 1，冷库节点到汇点的边的容量为第一问中所要求的那个比例，即$\left\lceil\frac{nb_i}{\sum b_i}\right\rceil$。然后运行网络流算法，如果求得的最大流值为 n，那么证明可以把所有集装箱在 1 小时内送到冷库，并且保证冷库的空间利用率绝对平均。如果小于 n，则不行。

评分标准：

1. 节点正确 +2 分
2. 边正确 +3 分
3. 容量正确 +3 分
4. 算法正确 +2 分



2. 如果无法保证冷库空间占用率绝对平均，如何设计最大流算法，判断能否保证集装箱在 1 小时内送到冷库？如果能，又如何尽可能的使冷库的空间占用率的最大值最小化（也就是使各冷库的空间占用率尽可能地平均）？ （10 分）

答案：对第一问，把最后一层边（冷库到汇点）的权重设为每个冷库的能存储的集装箱数量b_i，然后再运行最大流算法，如果求得的流值为 n，那么证明可以把所有集装箱在 1 小时内送到冷库。否则，就不行。

对第二问，可以先按第一小题中的方法设定每个冷库的空间占有率，如果求得的最大流小于 n 的话，那么用二分法从当前比例到 100%之间进行二分查找，找到可以使最大流值等于 n 的最小比例。如果第一问（即不要求冷库空间占用率平均的情况下）的回答是 Yes，那么这样的比例肯定是可以找得到的。或者也可以从当前比例开始，同比例地增加最后一层边的容量，直到最大流值达到 n 为止。

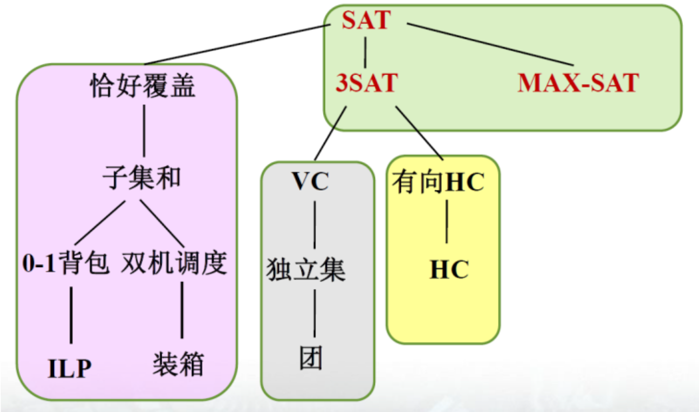
评分标准：
第一问：边的容量修改正确得+3 分，算法正确+2 分
第二问：算法描述清楚+5 分

得分

六、NPC 证明（15 分）

给定图$G=(V, E)$和整数k。如果任意两个结点$v, u \in I$，边$(v, u) \notin E$，并且也没有从v到u的两条边的路径，即没有结点w使得$(v,w) \in E \wedge (w,u) \in E$，则称集合$I \subseteq V$是强独立的。强独立集问题是要确定$G$是否有一个大小不小于$k$的强独立集。

1. （5 分）请证明强独立集是 NP。
2. （10 分）请证明强独立集是 NP 难。（提示：可以利用下图中的 NPC 问题进行证明）



参考答案：

1. 任意给定一个强独立集猜想，解释验证这个猜想只需要多项式时间，得满分。如果没有说明多项式时间，扣 2 分。
2. 证明一：说明独立集是强独立集的子问题，得满分。
   证明二：

利用独立集问题进行证明。

任给一个独立集问题的实例图 G，在每条边的中间加一个顶点 u_i，所有新加的顶点之间都连边，{u_i}形成完全图。然后证明在新的图 G’中，一个强独立集不能同时选择两个 u’，因为任何两个 u’有边相连。也不能同时选择一个新顶点和一个旧顶点，因为任意一个新顶点距离任意一个旧顶点距离为 2，所以只能都选旧顶点。新图中只由旧顶点构成的强独立集等价于旧图中的独立集。

评分标准：

选择独立集+2 分；增加节点+2 分；增加边+2 分；说明等价性成立+4 分。

只加点不加边得 6 分。

证明三：利用独立集问题进行证明。

任给一个独立集问题的实例图 G，在每条边的中间加一个顶点 u_i，所有新加的顶点之间都和一个公共顶点 s 相连，并且 s 与另外一个顶点 s’相连。也可以证明二者的等价性。和证明二类似。

得分

七、近似算法（10 分）

下列算法是解决最小顶点覆盖问题（VC）的一个近似算法，给定图 G=<V, E>，输出集合 C 代表找到的近似解。

```
Approx-VC (G)
1.  C 初始化为空集
2.  E1 = E
3.  While E1 不为空
4.      Do  任取 E1 中的一条边(u,v)
5.          C = C ∪ {u, v}
6.          从 E1 中删除所有和 u 或 v 关联的边
7.  Return C
```

（1） 证明算法的输出的确是 G 的一个顶点覆盖。（3 分）

答案：首先 C 是一个顶点覆盖，因为算法结束时 E 中每一条边都至少有一个顶点属于 C。

（2） 请分析该近似算法的近似比。（7 分）

答案：假设 A 是算法第 4 行所选中的边的集合，C*是最优顶点覆盖。由于算法中一旦某条边被选中，其他所有相邻的边都会被删除。所以，A 中任意两条边都不相邻（共享一个顶点）。因此，为了覆盖 A 中的每一条边， C*至少要有|A|个顶点，即 |C*|>=|A|。同时，算法每次选中一条边，C 就会增加两个顶点。所以，|C|=2|A|。从而，|C|=2|A|<=2|C*|。近似比为 2。

评分标准：证明 A 中任意两条边不相邻+2 分，证明 C*至少有|A|个顶点+2 分，证明|C|=2|A| +2 分， 证明近似比为 2 得 1 分。

得分

八、（15 分）

1. （5 分）在 n 个数中找最大数至少比较 n-1 次，如果找一个至少第$\lceil n/2 \rceil$大的数，至少需要多少次比较？

2. （5 分）如果想用更少的比较次数找出一个至少第$\lceil n/2 \rceil$大的数，就需要容忍一定的错误，请设计一个蒙特卡罗随机算法，使得在错误概率期望不超过 1/n 的情况下，找出一个至少第$\lceil n/2 \rceil$大的数。

3. （5 分）该算法的时间复杂度是多少？

（1） 至少需要比较$\lceil n/2 \rceil$次；（或取整错误，扣 2 分）

（2） 随机独立可重复的抽选$\lceil \log n \rceil$个数，每个数都不是至少$\lceil n/2 \rceil$大的概率小于 1/2，所有的数都不是至少$\lceil n/2 \rceil$大的概率小于$\left(\frac{1}{2}\right)^{\lceil \log n \rceil} < \frac{1}{n}$;

（3） 时间复杂度为$\lceil \log n \rceil - 1$或$O(\log n)$

（取整错误扣 1 分）