1.1 欧几里德算法的输入大小为 $\log_2 a$, $\log_2 b$,分别将算法中进行的求余操作和赋值操作的次数表示成 $\log_2 a$, $\log_2 b$ 的函数，并由此得出欧几里德算法的渐近复杂性。

1.2 理解下面的插入排序算法，并完成后面的分析。

插入排序算法 InsertSort

输入：数组 $A[1:n]$

输出：排序后的数组 $A[1:n]$

```
1. For   i←2   To   n   Do
2.       key←A[i];
3.       j←i-1;
4.       While   j>0 且 A[j]>key   Do
5.                A[j+1] ← A[j];
6.                j←j-1;
7.       A[j+1] ←key;
```

(a)证明算法必然停止；

(a)利用循环不变量方法，证明算法的正确性。

(b)分别分析最坏情况下、最好情况下、平均情况下算法执行的比较操作次数和赋值操作次数，将分析结果表示成 n 的函数。分析平均复杂度是，假设所有输入服从均匀分布。

1.4 考虑如下的素数判定算法，将整除判定操作视为基本操作。

素数判定算法 IsPrime

输入：输入正整数 n

输出：n 是否为素数

```
1. For   i←2   To   n^(1/2)   Do
2.       If   i 整除 n   then   返回“no”;
3. 返回“Yes”
```

指出算法的输入规模，将基本操作个数表达成输入规模的函数，指出这个算法是多项式时间算法还是指数时间算法。

1.5 考虑如下的计算斐波那契数列第 n 项的算法，将加法操作视为基本操作。

斐波那契 DP 算法

输入：正整数 n

输出：斐波那契数列第 n 项

```
1. If     n=0  或 1   Then  返回 1;
2. For   i←2   To   n
3.       F[i] ← F[i-1] + F[i-2];
4. 返回 F[n]
```

指出算法的输入规模，将基本操作个数表达成输入规模的函数，指出这个算法是多项式时间算法还是指数时间算法。

1) 证明或否证： $f(n)+o(f(n))=\Theta(f(n))$
2) 试证明：$O(f(x))+O(g(x))=O(max(f(x),g(x)))$.
3) 证明或给出反例：$\theta(f(n))\cap o(f(n))=\varnothing$。
4) 证明：设 k 是任意常数正整数，则 $\log^k n = o(n)$
5) 用迭代法解方程 $T(n)=T(9n/10)+n$.
6) 解方程 $T(n)=6T(n/3)+\log n$.
7) 解方程 $T(n)=3T(n/3+5)+n/2$.
8) 解方程 $T(n)=T(n/2)+1$.
9) 解方程 ：$T(n)=9T(n/3)+n$；
10) 解方程：$T(n)=T(n/2)+n^3$;
11) 解方程： $T(n)=2T(\sqrt[4]{n})+(\log_2 n)^2$；
12) 解方程：$T(n)\le\begin{cases}C_1 & n<20\\ C_2 n+4T(n/5) & n\ge 20\end{cases}$;

作业书写注意事项

1. **规范书写算法是交流算法的重要手段，是评分的重要依据；**
2. **算法应先分析问题特征、再表述算法思想、然后写伪代码、最后是分析；**
3. **算法伪代码不要依赖于具体的程序设计语言（参照课件中的例子）**

第 3 章习题

(1-11 题为必做题，11-16 选做两题即可)

3.1 根据表达式 $a^{2k}=a^k\cdot a^k$ 和 $a^{2k+1}=a^k\cdot a^k\cdot a$ 设计分治(或递归)算法求解下列问题，并分析算法的时间复杂度。

(a)输入实数 a 和自然数 n,输出实数 a^n;

(b)输入实数矩阵 A 和自然数 n，输出实数矩阵 A^n.

3.2 斐波那契数列满足递归方程 $F(n+2)=F(n+1)+F(n)$, 其中 $F(0)=0$, $F(1)=1$。

(a)分别用数学归纳法和第 2 章例 19 的结论，证明：

(1) $F(n+2)=1+\sum_{i=0}^{n}F(n)$;　　(2) $F(n+2)\geq\left(\frac{1+\sqrt{5}}{2}\right)^n$;

(b)设计算法根据递归方程计算 $F(n)$，将时间复杂度表达成 n 的函数;

(c)根据第 2 章例 19 中 $F(n)$的解析表达式，设计算法计算 $F(n)$，将时间复杂度复杂性表达成 n 的函数，并指明计算机运行该算法时可能遇到的问题。

(d)根据表达式$\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}F(n-1)\\ F(n-2)\end{pmatrix}=\begin{pmatrix}F(n)\\ F(n-1)\end{pmatrix}$和习题 3.1(b)的算法，设计算法计算 $F(n)$，将时间复杂度复杂性表达成 n 的函数。

(e)比较(b),(c),(d)得到的算法。

3.4 给定平面上 n 个点构成的集合 S，试设计一个分治算法输出 S 的三个点，使得以这三个点为顶点的三角形的周长达到最小值。(提示：模仿最邻近点的分治过程)

3.5 给定平面上 n 个白点和 n 个黑点，试设计一个分治算法将每个白点与一个黑点相连，使得所有连线互不相交，分析算法的时间复杂度。（提示：划分时类似于 GrahamScan 算法考虑极角，确保子问题比较均匀）

3.6 给定凸多边形 $p_1,p_2,\ldots,p_n$（边界逆时针顺序）和 n 个点 $q_1,q_2,\ldots,q_n$，试设计一个分治算法计算 $q_1,q_2,\ldots,q_n$ 中位于凸多边形 $p_1,p_2,\ldots,p_n$ 内部的点的个数，使其时间复杂度是 n^2 的严格低阶函数。

3.7 设 $X[0:n-1]$和 $Y[0:n-1]$为两个数组，每个数组中的 n 个均已经排好序，试设计一个 O(log n)的算法，找出 X 和 Y 中 $2n$ 个数的中位数，并进行复杂性分析。

3.8 设 $A[1:n]$是由不同实数组成的数组，如果 $i<j$ 且 $A[i]>A[j]$，则称实数对 $(A[i],A[j])$是该数组的一个反序。如，若 $A=[3,5,2,4]$，则该数组存在 3 个反序(3,2)、(5,2)和(5,4)。反序的个数可以用来衡量一个数组的无序程度。设计一个分治算法（要求时间复杂度严格低于 n^2），计算给定数组的反序个数。

3.9 给定一个由 n 个实数构成的集合 S 和另一个实数 x，判断 S 中是否有两个元

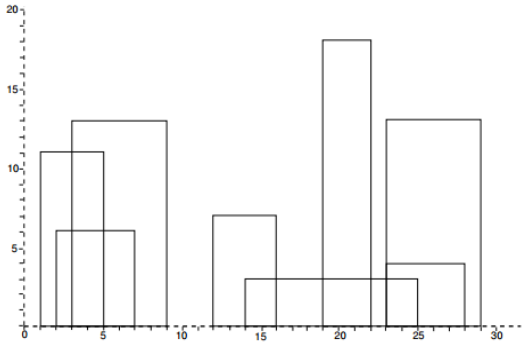
素的和为 x。试设计一个分治算法求解上述问题，并分析算法的时间复杂度。

3.10 设单调递增有序数组A中的元素被循环右移了k个位置，如<5; 15; 27; 29;35; 42>被循环右移两个位置（k = 2）得到<27; 29; 35; 42; 5; 15>。

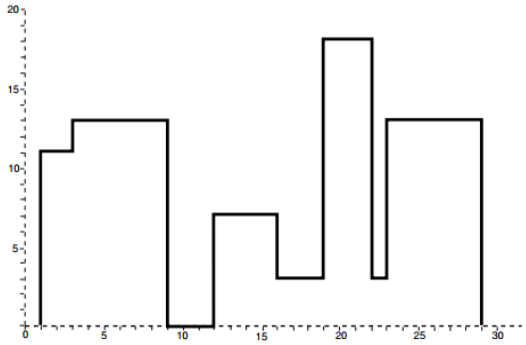
(1). 假设k已知，给出一个时间复杂度为$O(1)$的算法找出A中的最大元素。

(2). 假设 k 未知，设计一个算法在 O(log$n$)时间内找出 A 中的最大元素。

3.11 设建筑表示为三元组(x_L, H, x_R)，其中 x_L,x_R 表示建筑左、右两侧的 x 坐标,H 表示建筑的高度。如下左侧图中的建筑可以表示为三元组序列{(**3**, 13, **9**),(**1**, 11, **5**),(**12**, 7, **16**),(**14**, 3, **25**),(**19**, 18, **22**),(**2**, 6, **7**),(**23**, 13, **29**),(**23**, 4, **28**)。建筑群的 skyline 是一系列横坐标以及连接相邻横坐标的水平线高度来表示，下图右侧的 skyline 表示为整数序列(**1**, 11, **3**, 13, **9**, 0, **12**, 7, **16**, 3, **19**, 18, **22**, 3, **23**, 13, **29**, 0)。考虑设计分治算法求解如下计算问题：**问题定义**：**输入**：n 幢建筑构成的建筑群；**输出**：n 幢建筑的 skyline;



buildings



skyline

后面为选做题（选做两题即可）

3.12 给定平面上 n 个点构成的集合 S={p_1,…,p_n}。如果存在边平行于坐标轴的矩形仅包含 S 中的两个点 p_i 和 p_j（1≤i,j≤n）,则称 p_i 和 p_j 为“友谊点对”。试设计一个分治算法统计 S 中友谊点对的个数。

3.13 输入含有 n 个顶点的加权树 T 和实数τ，树 T 中每条边的权值均非负，树中顶点 x,y 的距离 dis(x,y)定义为从 x 到 y 的路径上各边权值之和。试设计一个分治算法输出满足 dis(x,y)≤τ的顶点对的个数。

3.14 给定非负数组 A[1:n]，B[1:m]和整数 k。试设计一个算法计算集合{A[i]·B[j]|1≤i≤n,1≤j≤m}中的第 k 小的元素。

3.15 设 M 是一个 m×n 的矩阵，其中每行的元素从左到右单增有序，每列的元素从上到下单增有序。给出一个分治算法计算出给定元素 x 在 M 中的位置或者表明 x 不在 M 中，分析算法的时间复杂性。

3.16 给定平面上 n 个白点和 n 个黑点，试设计一个算法判定是否存在一条不通过任何输入点的直线使得所有白点和所有黑点分别位于直线的两侧。（提示：对每个点利用划分寻找可能的直线，划分时类似于 GrahamScan 算法考虑极角）

第 4 章习题

4.1 设计动态规划算法输出数组 $A[0:n]$中的最长单调递增子序列。

4.2 考虑三个字符串 X,Y,Z 的最长公共子序列 $LCS(X,Y,Z)$。

(1)寻找反例 X,Y,Z 使得 $LCS(X,Y,Z)\neq LCS(X, LCS(Y,Z))$;

(2)设计动态规划算法计算 X,Y,Z 的最长公共子序列,分析算法的时间复杂度。

4.3 输入数组 $A[0:n]$和正实数 d,试设计一个动态规划算法输出 $A[0:n]$的一个最长子序列，使得子序列中相继元素之差的绝对值不超过 d。分析算法的时间复杂度。

4.4 给定一个整数序列 $a_1,\ldots,a_n$。相邻两个整数可以合并，合并两个整数的代价是这两个整数之和。通过不断合并最终可以将整个序列合并成一个整数，整个过程的总代价是每次合并操作代价之和。试设计一个动态规划算法给出 $a_1,\ldots,a_n$ 的一个合并方案使得该方案的总代价最大。

4.5 输入表达式 $a_1O_1\ a_2\ O_2\ldots\ldots O_{n-1}\ a_n$，其中 a_i 是整数$(1\le i\le n)$,$O_j\in\{+,\times\}(1\le i\le n-1)$。

(1)试设计一个动态规划算法，输出一个带括号的表达式（不改变操作数和操作符的次序），使得表达式的值达到最大，分析算法的时间复杂性。

(2)令 $O_j\in\{+,-,\times,\div\}$,重新完成(1)规定各项任务。

4.6 设平面上有一个 $m\times n$ 的网格，将左下角的网格点标记为（0,0）而右上角的网格点标记为（m,n）。某人想从（0,0）出发沿网格线行进到达（m,n），但是在网格点（i,j）处他只能向上行进或者向右行进，向上行进的代价为 a_{ij}（$a_{mj}=+\infty$），向右行进的代价是 b_{ij}（$b_{in}=+\infty$）。试设计一个动态规划算法，在这个网格中为该旅行者寻找一条代价最小的旅行路线。

4.7 集合划分问题描述如下：

输入：正整数集合 $S=\{a_1, a_2, a_3, \ldots, a_n\}$;

输出：是否存在 $A\subseteq S$ 使得 $\sum_{a_i\in A} a_i = \sum_{a_i\in S-A} a_i$;

试设计一个动态规划算法，求解集合划分问题。

4.8 输入凸 n 边形 $p_1,p_2,\ldots p_n$,其中顶点按凸多边形边界的逆时针序给出，多边形中不相邻顶点间的连线称为弦。试设计一个动态规划算法，将凸边形 $p_1,p_2,\ldots p_n$ 剖分成一些无公共区域的三角形，使得所有三角形的周长之和最小。

4.9 输入一棵加权树 T,其中每条边的权值均是实数，试设计一个动态规划算法输出权值最大的子树。

4.10 -4.13 选做两题即可

4.10 输入平面上 n 个点，点与点之间的距离定义为欧几里得距离。试设计一个动态规划算法输出一条先从左到右再从右到左的一条最短路径，使得每个输入点恰好被访问一次。

4.11 给定一个 $n\times n$ 的矩阵 A，矩阵中的元素只取 0 或者 1。设计一个动态规划算法，求解得到 A 中元素全是 1 的子方阵使其阶数达到最大值。

4.12 正整数 n 可以拆分成若干个正整数之和，考虑拆分方案的个数。

(1)令 $g(i,j)$表示拆分整数 i 时最大加数不超过 j 的方案个数，证明：$g(i,j)=g(i,j-$

1)+$g(i-j,j)$。

(2)根据（1）设计动态规划算法计算整数 n 的拆分方案个数，要求算法的时间复杂度为 $O(n^2)$。

4.13 设 $R(X)$表示将整数 X 的各个数位取逆序后得到的整数，如 $R(123)=321$, $R(120)=21$。现输入正整数 N，试设计一个动态规划算法计算方程 $R(X)+X=N$ 的解的个数，分析算法的时间复杂度。

第 5 章　习题

5.1 现有一台计算机，在某个时刻同时到达了 n 个任务。该计算机在同一时间只能　　处理一个任务，每个任务都必须被不间断地得到处理。该计算机处理这 n 个任务需要的时间分别为 $a_1,a_2,\ldots,a_n$。将第 i 个任务在调度策略中的结束时间记为 e_i。请设计一个贪心算法输出这 n 个任务的一个调度使得用户的平均等待时间 $1/n\sum e_i$ 达到最小。

5.2 现有面值为 1 角、5 分、2 分、1 分的硬币，每种硬币的个数都是无限的。给出一个贪心算法，　使得对任意给定的面值为 n（n>18）分的纸币能够将它兑换成币值相等的硬币且使用硬币个数最少。证明算法的正确性并分析其复杂度。

5.3 给定 k 个排好序的有序序列 $s_1, s_2, \ldots, s_k$,现在用 2 路归并排序算法对这些有序序列排序。假定用 2 路归并排序算法对长度分别为 m 和 n 的有序序列排序要用 $m+n$-1 次比较操作。设计一个贪心算法合并 $s_1, s_2, \ldots, s_k$ 使得所需的比较操作次数最少。

5.4 设计分治算法求解如下问题，并分析算法的复杂性。

输入：字符表 C={$c_1,\ldots,c_n$}的前缀编码 code(c_1),...,code(c_n)

输出：与前缀编码 code(c_1),...,code(c_n)对应的编码树 T

5.5 给定两个大小为 n 的正整数集合 A 和 B。对于 A 到 B 的一个一一映射 f，不妨设 $f(a_i)=b_i$（i=1,…,n），则 f 的代价为 $\sum_{i=1}^{n} a_i^{b_i}$ 。试设计一个贪心算法，找出从 A 到 B 的代价最大的一一映射。

5.6 一个 DNA 序列 X 是字符集{G, T, A,C}上的串，其上有大量信息冗余。设 x 是 X 的子串，x 及其冗余形式在 X 内在出现的起、止位置构成了一系列等长区间$[p_1,q_1],\ldots,[p_m,q_m]$。试设计一个贪心算法找出$[p_1,q_1],\ldots,[p_m,q_m]$中互不相交的区间的最大个数，即确定 x 的独立冗余度。

5.7 背包问题定义如下，输入背包容量 C 和 n 个物品,其中第 i 个物品（$1\le i\le n$）的重量为 w_i 且其价值为 v_i, 试设计一个贪心算法输出向量$\langle x_1,\ldots,x_n\rangle$使得 $0\le x_i\le 1$（$1\le i\le n$）且 $\sum_{i=1}^{n} x_i v_i$ 达到最大值。

5.8 给定平面点集 $P=\{(x_i,y_i)|\ 1\le i\le m\}$ 和 $Q=\{(x_j,y_j)|\ 1\le j\le n\}$。$(x_i,y_i)\in P$ 支配$(x_j,y_j)\in Q$ 当且仅当 $x_i\ge x_j$ 且 $y_i\ge y_j$。试设计一个贪心算法输出集合$\{(p,q)|p\in P,q\in Q,p$ 支配 $q\}$使得该集合中点对最多。

5.9 某工厂收到 n 个订单(a_i,b_i),其中 a_i 和 b_i 均是正整数$(1\le i\le n)$, 订单(a_i,b_i)希望在时间 b_i 之前获得 a_i 件产品。工厂的生产能力为每个时间单位生产 1 件产品。工厂希望拒绝最少数量的订单，并恰当地排序剩下的订单使得剩下的订单均能够被满足。试设计一个贪心算法求解上述问题。

5.10 输入 n 个区间$[a_i,b_i]$,其端点满足 $1\le a_i\le i\le b_i\le n$,试设计一个贪心算法选出最少区间覆盖$[1,n]$。

第六章 平摊分析 习题

1.在数据结构 D 上执行操作序列 $Op_1,...,Op_n$。当 $i=2^k$ 时，Op_i 的代价为 i；当 $i\neq 2^k$ 时, Op_i 的代价为 1。

(a)用聚集方法分析该操作序列的时间复杂度上界；

(b)用会计方法分析该操作序列的时间复杂度上界。

2 在二进制计数器上，除了 Increment 操作外，定义操作 Reset 的意义为将计数器的（所有二进制位）置 0。用数组实现一个 k 位二进制计数器，使得在初始值为 0 的计数器上执行由 Increment 和 Reset 构成的长度为 n 的操作序列的时间复杂度上界为 $O(n)$。

3 设势能函数ϕ使得$\phi(D_i)\geq \phi(D_0)$对所有 i 成立，但$\phi(D_0)\neq 0$。证明：存在势能函数ϕ'使得$\phi'(D_0)=0$ 且$\phi'(D_i)\geq 0$ 对所有 i 成立，并且用ϕ'和ϕ在每个操作上得到的平摊代价相同。

4 完成 6.2.3 节情形 3 的平摊代价计算。

5 有一个存储正整数的链表 A。操作 Add(A,x)的含义如下：如果 x 是奇数，则创建新结点存储 x 并添加到 A 的末尾；如果 x 是偶数，则创建新结点存储 x 并添加到 A 的末尾，再删除 A 中紧邻 x 的所有存储奇数的结点。用平摊分析方法分析，在初始为空的链表上连续执行 n 次 Add 操作的时间复杂度上界。

6 有序队列 Q 用于存储一系列元素，并支持如下两种基本操作 OrderedPush 和 Pop。OrderedPush(x)操作从 Q 的第一个元素开始删除所有小于 x 的所有元素，然后将 x 插入 Q 中作为第一个元素。Pop()操作删除并返回 Q 的第一个元素（如果 Q 为空，则返回空）。假设将有序队列实现为链表，证明：在初始为空的有序队列上执行长度为 n 的操作序列的平摊代价为 $O(1)$。

7 字典这种数据结构通常用来存储一系列元素，通常它需要支持元素插入 Insert 操作和元素查找 Lookup 操作。字典可以实现为有序数组，此时 Lookup 操作的代价为 $O(\log n)$而 Insert 操作的代价为 $O(n)$。字典也可以实现为链表，此时 Insert 操作的代价为 $O(1)$而 Lookup 操作的代价为 $O(n)$。字典也可以按如下更好的方式实现。

将字典实现为一系列数组，第 i 个数组的大小为 2^i,并且每个数组要么是空的要么是满的。每个数组都是有序的，但不同数组的元素之间的大小顺序是任意的。如果字典有 n 个数据项，则 n 的二进制表示给出了字典中不等于空的所有数组。例如，如果 n=11，由于$(11)_2$=1011,故字典中使用了 3 个非空数组，亦即第 1 个数组、第 2 个数组和第 4 个数组，而第 3 个数组为空；此时整个字典如下所示：

A[0][]={6}
A[1][]={2,13}
A[2][]=empty
A[3][]={1,6,7,8,10,15,17,29}

(a) 假设字典的 Lookup 操作实现为对每个非空数组执行二分查找。Lookup 操作的最坏时间复杂度是多少？

(b) 为上述实现方案设计 Insert 操作算法，分析其最坏时间复杂度。

(c) 在初始为空的字典上，连续执行 n 次 Insert 操作，其平摊代价等于多少？

第7章习题

7.1 现有两台计算机和 n 个计算任务，第 i 个任务在两台机器上的运行时间为 $A[i]$ 和 $B[i]$，此外还给出了所有计算任务的信息通讯代价(i,j,c_{ij}),其含义是：如果第 i 个任务和第 j 个任务分别运行于两台机器则这两个任务间信息通讯时间为 c_{ij}。试设计一个任务调度算法，使得所有任务被执行时的总时间最小。

7.2 有向图 $G=(V,E)$的路径覆盖是一个由路径构成的集合$\{p_1,\ldots,p_k\}$,其中路径 p_i 和 p_j（$i\neq j$）无公共顶点;路径覆盖中路径的条数 k 定义为路径覆盖的代价。试设计算法求解如下问题。

输入：无环有向图 $G=(V,E)$

输出：G 的最小路径覆盖

7.3 一个社团的成员正在安排暑假的度假计划，有旅行项目 $t_1, \ldots, t_n$ 可供选择，但参加 t_i 的人数以 c_i 为限。每人有若干个喜好的项目但是至多只能选择一个。推导出一个充要条件，使得（限制条件下）能够为各个成员安排其喜欢的项目。

7.4 设计一个算法求解下面的**相异代表系问题**或断言该问题无解。

输入：非空集合 $S_1,S_2,\ldots,S_n$

输出：输出元素 $x_1,x_2,\ldots,x_n$ 使得 $x_i\in S_i$ 对 $1\leq\forall i\leq n$ 成立。

7.5 *公交车驾驶员问题* 有 n 名公交驾驶员；n 条上午路线，其规定的行车时间分别是 $x_1, \ldots, x_n$；n 条下午路线，其规定的行车时间分别是 $y_1, \ldots, y_n$。如果一名司机在上午路线和下午路线中的行车时间总和超过 t，则可以获得加班费。目标是为每名司机分配一条上午路线和一条下午路线，使得总加班费最小。证明：对任意司机 i，为其分配第 i 长的上午路线和第 i 短的下午路线，将产生一个最优解。（提示：不要使用匈牙利算法；考虑矩阵的特殊结构）

7.6 计算机系给该系的 n 名学生开设 k 个讨论班，各讨论班的主题各不相同。每名学生参加一个讨论班；第 i 个讨论班可容纳 k_i 名学生且$\Sigma k_i = n$。每名学生提交了一个清单，将 k 个讨论班依其偏好程度排序。将学生分配到各讨论班，如果不存在两个学生使得交换他们的讨论班之后各自均找到了更喜欢的讨论班，则称该分配方案是稳定的。描述怎样用加权二分匹配来找出一个稳定的分配方案。

7.7 (1)一个农业公司有 n 个农场和 n 个加工厂，各农场生产的玉米均达到了一个加工厂的加工能力。将农场 i 生长的玉米送到加工厂 j 加工获得的利润是 $w_{i,j}$。试设计一个算法建立农场和加工厂的一一对应关系使得总利润最大。

(2)政府宣布玉米生产过量，并补贴农业公司使其不再生产玉米。如果农业公司不再使用农场 i，则政府给于补贴 u_i；如果农业公司不再使用加工厂 j，则政府给于补贴 v_j。如果 $u_i+v_j< w_{i,j}$，则农业公司使用边 x_iy_j 所获得的收益比政府提供的补贴总量高。为使得所有生产停止，政府支付的补贴必须对所有 i,j 满足 $u_i+v_j>w_{i,j}$。同时，试设计一个算法输出一个补贴方案，使得 $\sum u_i+\sum v_j$ 达到最小值。

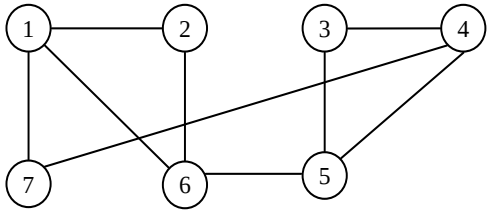
7.8 尝试如下改进 Ford-Fulkerson 算法。

(1)修改 Dijkstra 算法，使其在 G_f 中找到的增广路径是公差最大的增广路径，分析算法的复杂性；

(2)将 Ford-Fulkerson 算法中的增广路径替换成公差最大的增广路径，分析算法的复杂性。

第8章习题

1.在下图中考虑哈密顿环问题。将问题的解空间表示成树，并分别利用深度优先搜索和广度优 先搜索判定该图是否存在哈密顿环。



习题 1

2	3	
1	8	5
7	4	6

起始格局

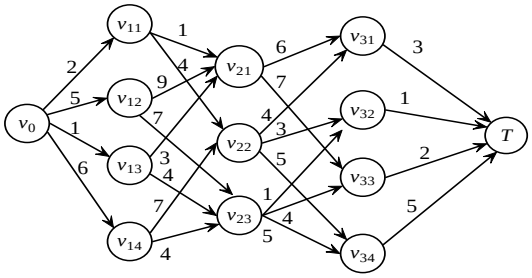
1	2	3
8		4
7	6	5

目标格局

习题 2

2.考虑 8-魔方问题。分别用深度优先方法，广度优先方法，爬山法，最佳优先方法判定上图所示的初始格局能够通过一系列操作转换成目标格局，将搜索过程的主要步骤书写清楚。

3. 分别用分支限界法和A^*算法求出下图中从v_0到T的最短路径，写出算法执行的主要过程。



习题 3

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} \infty & 5 & 61 & 34 & 12 \\ 57 & \infty & 43 & 20 & 7 \\ 39 & 42 & \infty & 8 & 21 \\ 6 & 50 & 42 & \infty & 8 \\ 41 & 26 & 10 & 35 & \infty \end{pmatrix} \end{array}$$

习题 4

4.在上面邻接矩阵给出的有向图上，用分支限界法求出代价最小的哈密顿环。

5.分别使用深度优先法和分支限界法求解子集和问题的如下实例。

输入：集合 S={7,4,6,13,20,8}和整数 K=18

输出：$S'\subseteq S$ 使得 S'中元素之和等于 K

6.精确描述求解 8-魔方问题的 A*算法，在习题 2 给出了起始格局和目标格局上给出 A*算法操作的主要步骤。

7.选用恰当的搜索策略，求解如下计算问题。

输入：正整数集合 $A=\{a_1,a_2,...,a_n\}$和正整数 K

输出：由 A 中任意 K 个数相加得到的最小素数

(1)将问题的解空间表示成一棵树；

(2)写出求解问题的通用算法；

(3)在问题实例 A={3,7,12,19}，K=3 上写出算法运行的主要过程。