算法

稳定匹配问题

首先定义稳定匹配的概念,所谓稳定匹配的情况是指没有任何一个人的伴侣有出轨的可能，因为别人对自己的伴侣的喜爱程度要更高。

Gale-Shapley算法的过程：男士根据顺序按自己对女士的好感度由高到底进行表白，若当前女士没有男友则表白成功；若当前女士有男友，如果在女士心中表白男士的好感度高于现任男友，则改变男友为表白男士；若好感度低于现任男友则表白失败，继续对下一位女士表白。直到所有栈中男士全都被pop掉。

算法正确性证明：反证法：假设GS算法配对的序列中有不稳定的情侣。即存在至少两对情侣 a-A，b-B中A喜欢b甚于a且b喜欢A甚于B。这种情况是不存在的，按题设情况在GS算法中b一定会先向a表白而且A一定会表白成功，所以不存在不稳定的匹配。

解的唯一性：存在解不唯一的情况，如

```
a：AB
b：BA
A：ba
B：ab
```

此时，有两种稳定的解：

```
a-A，b-B或a-B，b-A
```

伪代码:

```
while(!stack.empty())//从栈中弹出男生直到栈空为止
        for(auto a:boylist) //遍历该男生对女生的好感序列
        if(a.havenoboyfriend)
                match(a,nowboy);//如果该女生没有男友则配对成功
            stack.pop();//该男生被弹出栈
        if(a.haveboyfriend) //如果该女生有男友
                if(a.morelove(nowboy,oldboy)) //如果该女生更喜欢新男友
                match(a,nowboy);//配对成功
                stack.push(oldboy)//旧男友被压入栈中
            else//女生更喜欢现任男友
                continue;//男生根据序列继续向下一个女生表白
```

大学入学申请问题

设计一个英国大学入学申请程序。

分析问题和稳定匹配问题一样，所有使用稳定匹配问题求解即可。但有三种额外情况：

```
1.当校生数不同，一校一生时
        假如n<m，至多nm次循环，存在n-m个没收到offer。稳定，学校喜欢已录取的甚于没收到offer的。
2.当n<m并且一校多生的情况，每校不超过招生计划时。
        至多nm次循环，可能存在没收到offer的。
```

3.当有些人坚决不申请某些学校。
标记，遇到标记不进行匹配。

综合三种情况，依然可以解决大学申请入学问题。问题的求解是一个不断螺旋式上升的过程。

问题变换类型

复杂的问题转换为简单的问题
难解的问题转换为易解的问题

Case1 复杂的问题转换为简单的问题

实例化简：同一问题的更简单更便利的实例

预排序：无序数组变为有序数组

高斯消元法：将系数矩阵转为上三角矩阵

元素唯一性问题（重复元素问题）： n个元素的数组中每个元素是否都是唯一的。 预处理，排序后比较相邻两个元素是否相等。

Case2表达变换，同一实例的不同表示

医院实习安排——大学入学申请

堆 和 堆排序

查找 和 AVL

多项式计算（霍纳法则）

算法1，从左到右$O(n^2)$ 算法2，霍纳法则利用前面的结果$O(n)$

Case3将隐式问题转化为显示问题

约简到树，约简到图

3个传教士和3个野人过河每次两人如果两岸和船上传教士不能小于野人人数。

Case4未知的问题转化为已知的问题

等价问题

特殊与一般

（m，n）的最小公倍数=$m×n/gcd(m,n)$欧几里得

独立集（点之间互不直接连通），最小顶点覆盖的顶点是互补的。

最大团问题（任何顶点之间直接相连）：对于任一无向图$G=(V,E)$其补图$G1=(V1,E1)$定义为最大团等于最大独立集。

区间调度问题：N个报告，每个报告有开始时间和结束时间。 区间转换为点，区间冲突转换为边，然后求一个最大独立集。

算法分析

算法选择

同一个问题有多个模型，多个算法。

查找问题：从n个数中找某个值。 单次查找：多次查找：数据量大：

用以下5个符号来表示时间复杂度，O上界,Ω下界,θ上界下界相等,o小于,ω大于.

时空矛盾

时间资源称为时间复杂度，空间资源称为空间复杂度。

$O(n)$的计算规则

上界O和下界Ω的数量级是相同的。

- $ O(f)+O(g)=max(O(f),O(g)) $
- $ O(f)+O(g)=O(f+g) $ +$ O(f)O(g)=O(fg) $
- $ 如果g(N)=O(f(N)),则O(f)+O(g)=O(f) $
- $ O(Cf(N))=O(f(N)),其中C是一个正的常数 $
- $ f=O(f) $

O的阶越低越有价值,结果越精确; Ω的阶越高越有价值,结果越精确

有效算法

特性当输入规模加倍的时候,算法降低C倍C为一个常数.

如果一个算法是多项式时间算法,它是有效的。

复杂度比较

阶的高低

如果有多项式$f(n)=a_0+a_1n+\cdots+a_nn^d$

如果a>0,n>0,则有$f(n)<=\sum a_nn^d=O(a_nn^d)$ $f(n)>=\sum a_1n^0=O(a_1n^0)$

对数特性

$log_ab=\frac{log_b n}{log_ba}=Clog_bn$ 对数阶低于多项式阶

复杂度的比较方法

- 对数：取多个函数，比较大小得到现有函数的时间复杂度
- 积分：Stirling公式 $n!=\sqrt{2\pi n} (\frac{n}{e})^n(1+\Theta(\frac{1}{n}))$
- 极限：
- 放大

实例

- 常数阶:1,$n^{\frac{1}{logn}=O(1)}$,$logn^{\frac{1}{logn}}=logn*\frac{1}{logn}=1$
- 对数阶:log^2n,$logn$,$(logn)^{\frac{1}{2}}$,log^{logn}
- 多项式阶:n^3,$2^{logn}=n$
- 指数阶:$n!$,$n2^n$,2^{2^n},$(\frac{3}{2})^n$,$logn^{logn}=n^{log^{logn}}$

分析示例

非递归算法的时间复杂度分析

- 输入规模n
- 关键操作
- 最好最坏平均情况
- 计数关键操作次数

并列语句:复杂度相加

循环语句:循环体的运行时间*循环次数

子程序:子程序的复杂度*调用次数

- 常数阶:交换俩个变量
- 线性阶:从n个数中查找最大数,顺序查找
- 对数阶:
  - 从n个数中查找最大数,折半查找,分块查找
  - 求幂问题:快速幂原理

- 平方阶:
- 元素对枚举
- 矩阵
- 立方阶:不交集:给定n个集合S1...S2每个集合是{1,2...,n}的一个子集,是否存在不交子集
- 多项式阶:K独立集,是否存在k个顶点,彼此之间无边相连.
- 指数阶:独立集,给定图,求最大独立集,枚举所有子集

时空均衡

- 空间复杂性S(n):算法执行所需空间(储存器)资源量

以空间换时间:

- 预处理:对输入预处理,对获得 的额外信息存储,加速求解
- 预构造:简单使用额外空间实现更快的数据存储 散列法
- 动态规划:计算的子问题储存在表中,杜绝重复计算

散列法:使用链表,每一项对应链表平均长度是$\frac{n}{m}$,m是散列表长度.

平均探测数=$1+\frac{\alpha}{2}$,$\alpha=\frac{n}{m}$ 最坏情况依然是线性.

- 邻接矩阵:$O(n^2)$,无向图的邻接矩阵对角线对称,吴向图的列之和=列之和=顶点的度,有向图的行之和=出度 列之和等于入度
- 邻接表:空间是2m+n,有向图是m+n,检查(u,v)是否是边:$O(deg(n))$的时间,检查所有边$O(m+n)$,邻接表存储,可以表示重边和环.
- Breadth First Search:

算法设计技术

蛮力Brute-Force

然后就有很多很多

巧妙填数

只要枚举前三个数就可以了

百钱百鸡

手算方程非常简单，但是通过枚举的方法也可以解决。通过三个for循环可以判断，而且根据初始钱数的多少，可以算出每个变量的取值范围。进一步优化，可以只使用两个for循环来进行，只枚举公鸡和母鸡。

除法UVA725

- 减少枚举变量：n给定，枚举分母就可以知道分子的取值范围

分数拆分

- 减少枚举变量：

约瑟夫问题×

贪心算法的三个要素

- 最优子结构
- 贪心选择
- 无后效性

区间问题

区间调度问题

不相交区间问题,活动安排问题n个活动 $E=\{1,2,\cdots，n\}$ 都要求使用同一资源，且同一时间仅有一个活动可以使用该资源。求最大的相容活动子集，贪婪准则的选取：

- 最早优先开始
- ***最早结束优先***
- 区间最短优先
- 冲突最少优先

按结束时间来进行排序。

```
sort(A,a+n);

a[1]=true;
j=1;
for i=2 to n do {
        if(s[i]>=f[j])
        then a[i]=true
}
```

时间复杂度为 $O(n)$ 。

交换论证

把任意一个解逐渐变为贪心算法的解不会影响其最优性。只会变好不会变坏，反证法可以论证。

假设有多个报告厅的情况，目标使用最小的房间数安排下所有活动，此时采用的贪婪准则是：按开始时间。优先安排相容的时间。

```
sort(A,a+n);
d =0;
for j=1 to n do
        if(相容)
                then 加入
        else d=d+1
          安排到新房间
```

时间复杂度 $O(n logn)$

证明

证明最优的方法：给出每个解的界，证明算法的解正好等于这个界。

- 贪心性质+最优子结构
- 交换论证
- 结构性的界

区间选点 poj1328,3069

海岸线雷达覆盖岛屿。

以小岛为圆心画圆，以d为半径，得到与海岸线相交的区间 $[s_i,f_i]$ ，如果区间出现嵌套，则取最内部的区间。

所有小岛按终止位置排序，在第一个点终止位置安置雷达，去掉包含该点的区间，再选择第一个点

区间覆盖

数轴上n个闭区间 $[a_i,b_i]$ 最少的区间覆盖制定区间 $[s,t]$

贪婪准则：先切掉多余区间，根据起始时间排序，根据区间长度来进行排序

时间复杂度为 $O(n^2)$

最小生成树

无向联通赋权图 $G=(V,E)$ 包含所有定点的联通子图，且各边权最小。

环和割集的交集一定是偶数。

割的特性：设 S是最小顶点子集，c是正好一个定点在S中的边权最小的边，则最小生成树包含c。设所有边权cc都不同。prim算法。

$$ \begin{align*}

T =& \emptyset \ S =& {1}\ whi&le(S != V) do \ & 取边(i,j)\quad i \in S 且 j \in V-S 的最小边权，\quad 数组n\quad 堆logn\ & T = T \cup [i,j] \end{align*} $$ 环的特性：设C是G中的环，f是属于C的边权最大的边则最小生成树中不包含c。 克鲁斯卡尔：根据边权非递减排序

- case1：e加入t，产生环，按照环的特性，则不属于MST
- case2：e加入t，按照割的特性，属于MST

$$ \begin{align*} sort &(E)\ T =&(V,\varphi) 每个定点生成单定点子树\ S =&\ whi&le（E \quad and \quad T 中所含边数<n-1） do\ &从E中选取当前最小边,<u,v>\ &删除<u,v>\ &采用并查集判断 \end{align*} $$

并查集

find，union，每次合并，集合元素数加倍，合并操作的时间复杂度是 $nlogn$

哈夫曼编码

递归

斐波纳契数列