算法导论下半学期

图的算法

Kruskal：每次添加不构成环的最短路
Prim：以一个点为起点，每次增添所有节点中不构成环的最短路径

- 2-SAT问题转化为图
  对于可满足性有n哥变量组成m个clause

$$(x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3) \wedge (x_1 \vee \bar{x}_2) \wedge (x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_4)$$

- G_I has $2n$ nodes, one for each variable and its negation.
- G_I has $2m$ edges: for each clause $(\alpha \vee \beta)$ of I, G_I has an edge from the negation of α to β, and one from the negation of β to α.

如果x与非x组成强连通图就是不可满足的
如果没有上述情况，就是可满足的

Problem的性质

- **Tractable and Intractable Problems**
  Tractable：可以在多项式时间内解决的问题
  Intractabke：无法在多项式时间内解决的问题

- **Decision problem**
  Decision problem是只有 YES 或者 NO 的问题

1. 元素唯一性
2. 序列中有几个元素
3. 图中是否含有大小为k的clique
4. 图的最大连通分量（clique）
5. 图着色问题，给定m个颜色，是否让相同颜色的不相邻
6. 把图尽量染成少的颜色，会有几个颜色（chromatic number）

- **Class P**
  Deterministic：算法每一步只有一个选择，输入相同的input，输出永远不变
  class P：可以用deterministic的算法解决的decision problem
  如：2-coloring，2-sat，2-dm
  Class P具有封闭性
- **Class NP**

nondeterministic：可能有guess或vertification的内容（猜测与验证）比如找大质数、找公约数
class NP：可以用nondeterministic的算法解决的decision problem
如：coloring问题， 算法给出猜测，再给出验证

- **P与NP**
  P：一个问题可以在多项式（O(n^k)）的时间复杂度内解决。
  NP：一个问题的解可以在多项式的时间内被验证。

- **polynomial time reduction多项式时间归约**
  归约的意思是为了解决问题A，先将问题A归约为另一个问题B，解决问题B同时也间接解决了问题A。
  A与A′是两个decision的problem：

1. 如果问题A可以通过多项式时间的基本运算步骤转换为问题A′
2. 问题A′可以在多项式时间内被求解

这样，A reduce to A′ in polynomial time

在NP以前，未知问题归约到已知问题，去解决
在NP以后，已知的问题归约到未知的问题，去证明

- **NP-Complete NP完全问题**
  NP-hard：任意np问题都可以在多项式时间内归约为该问题，但该问题本身不一定是NP问题。
  NP-complete：既是NP问题，也是NP-hard问题。
  可满足问题就是NP-C问题

  如果是NP问题 也 由NPC问题归约到 则可以证明该问题也是NPC问题

- **4 color 归约到 7 color 问题**
  假设有7colorable的算法，相连即可
  原图上加三个点，三个彼此有边，所有的点都与这三个点相连，则4colorable变成7colorable， 则4color归约到 7 color

- **汉密尔顿环 归约到 汉密尔顿路径**
  汉密尔顿路径上s与t之间加一个点，如果形成汉密尔顿环，必经过新增加的点，则汉密尔顿路径为删除该点的路径

- **汉密尔顿问题 归约到 TSP（traveling salesman）问题**
  把汉密尔顿环的边赋值为1，变成完全图新增的边赋值为很大的值， 看是否有小于k的值，若不存在一定用了新的边；若存在则用的都是旧的边，存在汉密尔顿环

- **NPC问题的传递——clique、vertex cover、独立集问题的相互归约**
  A归约B，B归约C则A归约C
  如果A、B是NP问题且B是NP-C，那么

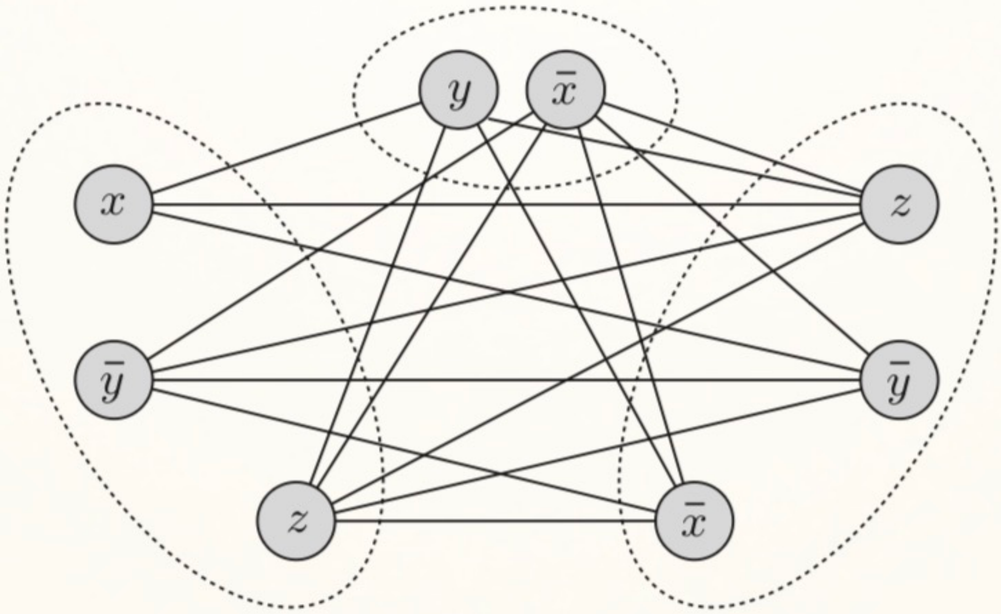
  Vertex cover （最小顶点覆盖）问题：
  图G的顶点覆盖是一个顶点集合V，使得G中的每一条边都接触V中的至少一个顶点。我们称集合V覆盖了G的边。最小顶点覆盖是用最少的顶点来覆盖所有的边。顶点覆盖数是最小顶点覆盖的大小。
  Independece set（独立集）问题：
  独立集是指图 G 中两两互不相邻的顶点构成的集合。
  clique归约到independence set
  在原图中是clique，在补图中就是一个independece set
  vertex cover归约到independence set
  给定一个图，如果v'是vertex cover，则v-v'是independence set
  反证：如果有两个v-v'的点没被cover掉，则矛盾

  以上三种问题是相同的问题

- **可满足问题 归约 clique问题**
  假设可满足问题f有n个bool variable 组成m个clause
  那么设置图G（所有occurance都有自己的点），如果有大小为m的clique，则f可满足

  同一个clause不同literal没边，不同clausex于非x没边，则找一个m大小的clique在每个clause里各取一个

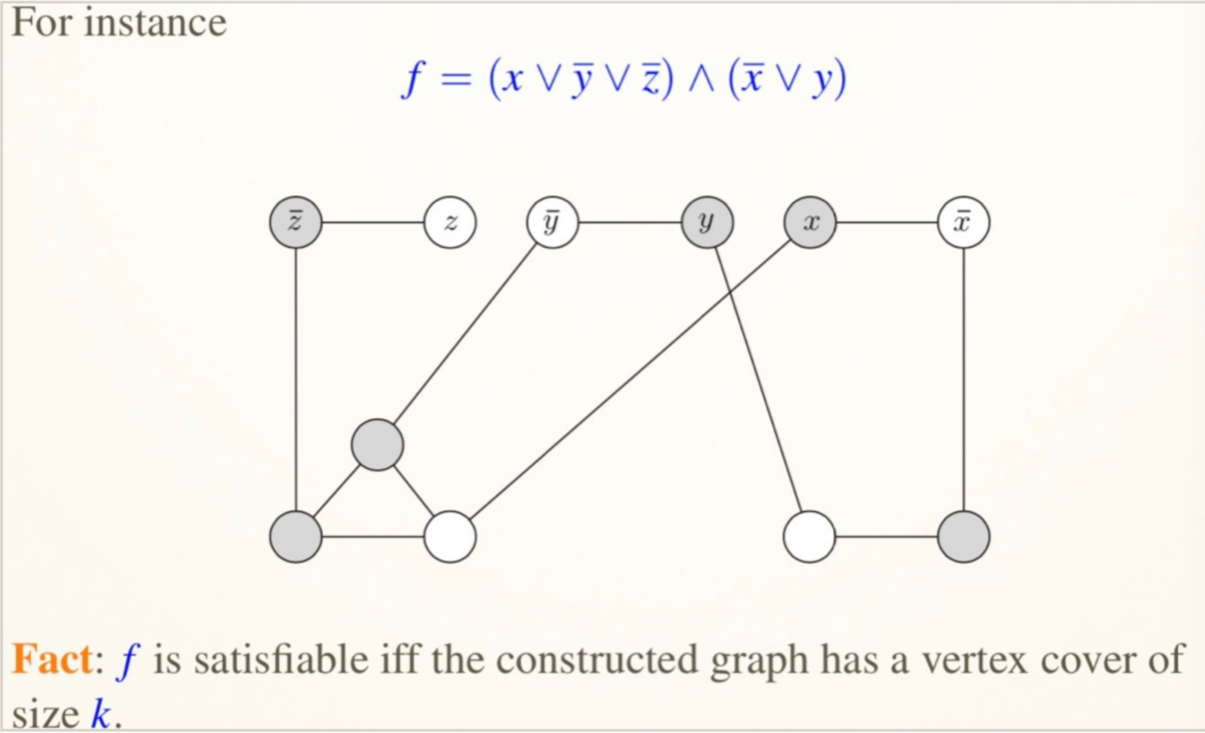
$$f = (x \vee \bar{y} \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y} \vee z)$$



- **可满足问题 归约 Vertex cover 问题**

  假设可满足问题f有n个bool variable 组成m个clause

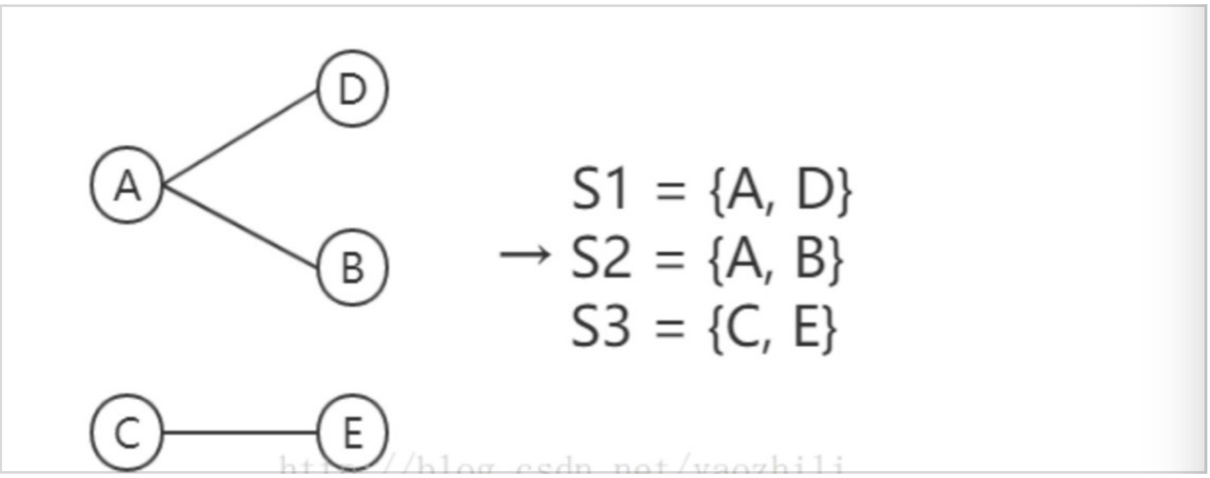
1. 对于每一个变量xi，都有链接xi与非xi
2. 每个含有不同nj的literal， G有大小相当的clique
3. Cj中的每个的点w，都有一条链接边和链接对应的点
4. Let $k = n + \sum_{j=1}^{m} (n_j - 1)$.



- **Vertex Cover 归约 Independence set独立集问题**
  Let G = (V,E) 是无向连通图. Then S ⊆ V is an independence set iff V \ S is a vertex cover in G.

- **NP-C问题汇总**
  3-SAT
  3-Coloring
  3-DimensionalMatching
  hamiltonian path（正好每个节点访问一次）
  Longest path两个定点之间是否有大于或等于某长度的一段路
  Partition 把一个集合分解成两个，两个集合的和相同
  Binpacking给定n个物体，bin容量为C，是否可能用k个bin装n个物体
  Setcover 给定一个集合X，F是x的子集， 是否存在k个subset，并集是x
  Knapack给定n个物体有重量与价值，knapsack容量C，能否装价值超过k的

- **Vertex Cover 归约 Set Cover**
  对于顶点覆盖问题的图G，对G中的每一条边创建一个集合Si，Si的元素为该边的两个顶点。如下图所示，将G归约到一组集合S1，S2，S3：



只需要选取碰撞集的解H对应的所有点，即为对应顶点覆盖问题的解。比如对于图中的碰撞集问题，假设b为2，则一个解为H={A，C}，此时，选取A和C点构成顶点集合V，即为顶点覆盖问题的解。
碰撞集无解时，顶点覆盖问题无解。用该命题的逆否命题证明，当顶点覆盖问题有解时，用对应的解V中的每个顶点对应于生成的那一组集合的元素得到的集合H，即为碰撞集问题的解。比如对于图中的的顶点覆盖题，其中一个解为V={A，C}，则碰撞集的一个解为H={A，C}。

vertex cover是set cover的子集，要求frequency是

- **SAT 归约 3-SAT**

  如果一个SAT每个子句的长度都为3，则就是3-SAT问题，只需要考虑大于3的，等价于3: 找n-2个变量：

  a1 ∪ a2 ∪ a3……∪ an

  (a1 ∪ a2 ∪ x1）(非x1 ∪ a3 x2) …… (非xn-2 an-1 an)

  上下的字句可满足性等价

  一个变量可以用一大堆变量表示

  (非xn ∪ x1）(非x2 ∪ x3) ……(非xn-1 ∪ xn)

  同一个变量会出现很多次，

  希望硬件固定：每个文字只出现3次：

  新加入的变量(非xn ∪ x1）(非x2 ∪ x3) ……(非xn-1 ∪ xn)代替出现4次以上的，最后再加入这个这一串，使x1 x2……x3相同

- **Class co-NP**

  Co-Np包含那些complement是NP的

- **complete for co-MP**

A是co-Np 且 每一个co-NP的问题都可以归约到A

Co-NP问题汇总：

A是NP-C问题 当且仅当 A的complement A'是co-NP-C问题

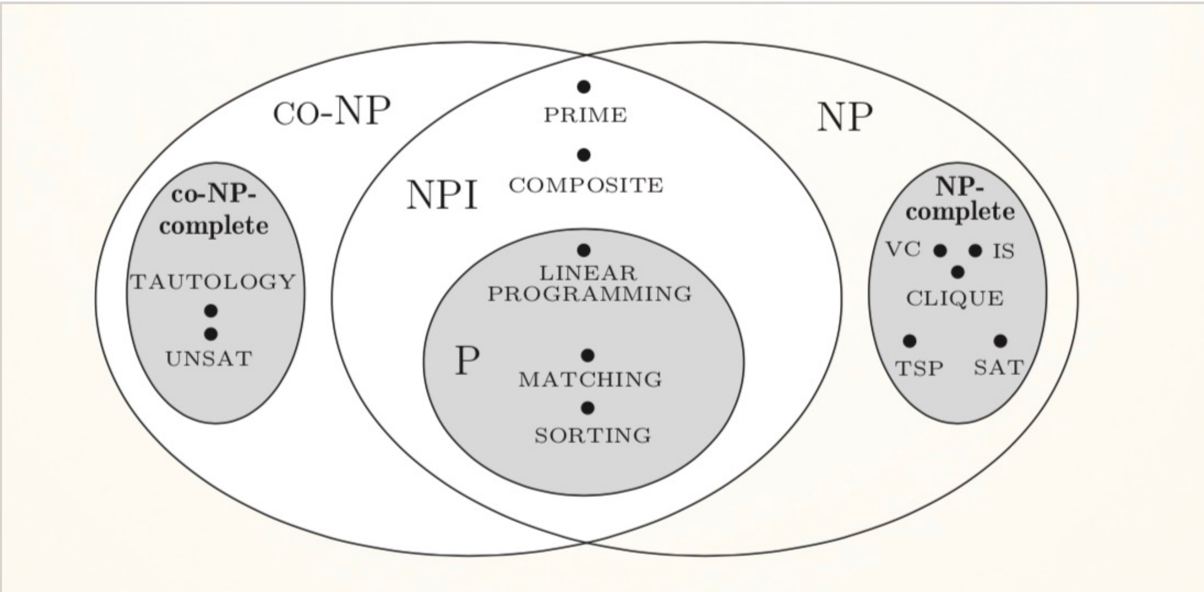
1. unsat问题
2. tautology问题

- **Class NPI**

  如果问题A和它的补A' 都是Np-C的，那么 co-NP=NP

  事实： 如果 co-NP≠NP then NP≠P

  NPI是co-NP与NP的交集（也就是一个问题和他的补集都是NP问题，也都是co-NP问题）



prime 与composite已经在p中了

图同构问题还在npi中

运算的复杂度

addition O（n）

mult O（n^2）-）（n^3/2）

Mod. O(n)

指数运算 O(n^3)

欧几里得求最大公约数 O（n^3）

Inverse O(n^3)

- **modular inverse**

If gcd(a,N)>1, then ax!=1 mod N

证明：ax mod N = ax + kN

Gcd(a,N)整除ax mod N

- **modular division**

密码学

- **费马小定理**

If pis a prime, then for every 1 ≤ a< p,

a^p−1 ≡ 1 (mod p)

证明：

a^p-1 (p-1)! ≡ (p-1)! Mod P

因为是素数，所以!=0

如果是素数 可以通过费马测试

但不意味着如果不是 就通不过

如果是合数，希望大部分的都通不过费马测试

卡米可儿数： 所有小于的数字都可以通过测试

561 = 3*1 1 *17

卡米可儿数是无穷的

在没有卡米可儿数：

如果是素数 所有都可以通过

合数： 大部分无法通过

如果有一个无法通过，则至少有一半无法通过，

假设a通不过 a^N-1 !=1 mod N

b^ N-1 = 1 mod N

那么如果a是素数 都能通过

如果a是合数，则至少有一半无法通过

- **生成随机素数**
  随机产生一个数， 测他是不是素数
  拉格朗日素数定理： 素数产生的概率π(x) ≈ xln(x)

- **Cryptography**
  eavesdropper 窃听
  Intruder 假装是一个人 把信息发给他

  两个素数p、q
  令N = pq
  找到一个e 与（p-1）（q-1）互素
  E在p-1与q-1中有一个模倒数
  d是一个模倒数

  (x^e)^d≡ x (mod N)

线性规划

目标函数是可解空间中的一组平行线
是否最优解都在顶点取到
两种情况不等式使得顶点不能取到要求的内容：

1. infeasible不等式构成不存在的空间
2. runboundedegion太宽松了以致于是无边界的

线性不等式 是一个平面

- **simplex method**
  从一个顶点出发 去看相邻顶点，如果相邻顶点值更优，移到该顶点，依次这么做，找到最优的点 → 到达最优的点

  M个不等式 n个变量：
  需要n个方程取等号 C（n,M）

各有n个等式构成的顶点，其中有n-1个相同的等式的两个顶点就是相邻顶点
Simplex是一个指数级的算法，却很高效：smooth analysis 分析

- **椭球法**

- **最短路径归约到线性规划 shortest path to LP**

duality 对偶

在左面找一个式子，最小化最大值

对偶线性规划：找到最小的最优解的线性规划

Primal LP:

$$
\begin{aligned}
\max\ c_1x_1 + \cdots + c_nx_n& \\
a_{i1}x_1 + \cdots + a_{in}x_n \le b_i \quad &\text{for } i \in I \\
a_{i1}x_1 + \cdots + a_{in}x_n = b_i \quad &\text{for } i \in E \\
x_j \ge 0 \quad &\text{for } j \in N
\end{aligned}
$$

Dual LP:

$$
\begin{aligned}
\min\ b_1y_1 + \cdots + b_my_m& \\
a_{1j}y_1 + \cdots + a_{mj}y_m \ge c_j \quad &\text{for } j \in N \\
a_{1j}y_1 + \cdots + a_{mj}y_m = c_j \quad &\text{for } j \notin N \\
y_i \ge 0 \quad &\text{for } i \in I
\end{aligned}
$$

考试：给定一个等式，求对偶

- **Complementary Slackness**

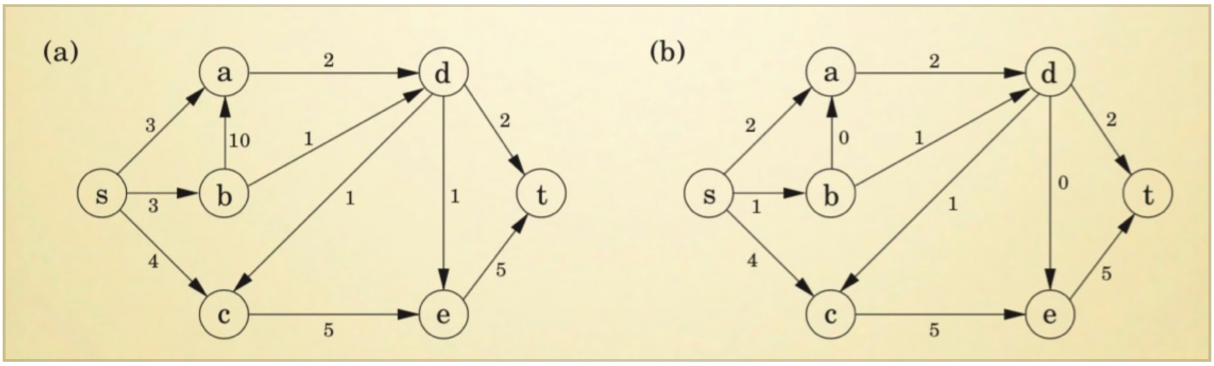
给定一个问题知道n个不等号取等号取最优值

如果对偶中最优解的变量=0， 那么原等式取不到， 如果对偶中的最优解的变量>0， 那么原等式取等号

Flows in Networks

运油问题： 用network输油， 目标：从源头到目的地越多越好
每一个pipeline都有一个最大的容量

假设（a）容量图无环， 右侧（b）中流量小于等于容量，每个节点入的流量之和等于出的流量之和



每一条边一个ce表示最大的

三个条件：
流量小于容量
每个中间过程中的节点的进=出
流量不为负
求：源头或目的地的流入或流出量的最大值

- **一个等式可以变成两个不等式**
  一条边上没有容量限制，从目的地到起始地
  这样：
  从起始点开始， 流入的量小于等于流出的量

- **剩余网络容量算法**
  根据几何方式找临接点，从原点到终点找一条路径，有一个容量最小的路径，

  若一条路径是原网络还没到达最大的，新的容量就是没用完的容量
  如果周围都是flow，容量= 流量，就加一条返向边，返回的容量就是他的流量

- **最小割**
  找到一个cut把源头和终点分割， 把割的容量相加

  如何算出最小割：
  容量= 流量的时候边不见了， 容量=流量 （出现反向边）

近似算法

每个instance有一个solution，有一个最大值或最小值，还有一个最优值
通过一个天然的关系设计算法
希望找|A-OPT|<k，绝大多数不存在这样的近似

- **bin packing（NPC）**

给一堆物品0-1的大小，一堆大小为1的箱子，最少使用多少个箱子装物品

1. FF（first fit）

   随机拿物品，一次尝试箱子是否能装得下

   FF算法是一个近似比为2的算法

   FF<=2*OPT

   天然的关系：所有物品加起来<=OPT

   每一个FF方法得到的箱子都是半满的

   那么FF-1<=2*所有物品加起来

   这种问题数字越大越近似，数字越小越不近似

   一定不存在FF/OPT<=3/2 -x的算法， 因为如果如此，OPT=2，那么近似算法得到的OPT也是2。因此当数量很小的时候，无法得到近似算法。

2. BF（best fit）

   找到最合适装的，装完该物品剩余的空间最小

- **欧几里得TSP（travel sales person）问题**

  一个完全图，满足两边之和>=第三边的问题

  最小生成树是类似的算法：每个点都连起来 总权值最小

  欧几里得TSP的近似算法：

1. 在图上找最小生成树
2. 最小生成树上所有的边double，一定是一个欧拉图（度=偶数）
3. 在边上找一个欧拉洄游
4. 在欧拉洄游的任何一点开始顺着欧拉洄游走，当那个点曾被访问，跳到下一个点，保证每个点不重复

   近似比=2：

   天然的关系：MST<= OPT -1 <= DOPT

   最终解A<=欧拉图 = 2MST<=. 2(OT-1)<2OPT

   改进：把最小生成树的奇数度找出来，找一个match，补到树上，构成一个欧拉图，在那个欧拉图上找近路

   天然的关系： MST<=OPT

   match<=1/2* OPT

   Match点连起来 M+M’<= OPT

   M是最小的，因此M<=1/2OPT

   R<=3/2

- **general 的 TSP问题**

  如果有任何近似比的算法<无穷，则P=NP

  汉密尔顿图 ➜ TSP

汉密尔顿图不一定是完全图，边上没权值 ➡ 补齐 ➡ 原边1，补的边N ➡ 是否有大小为n的图，则走完所有的图

如果解是100， 近似算法找 <= Ra*100
如果存在近似算法，则可以精确地出汉密尔顿图的近似算法

- **vertex cover问题**
  找最大match，随便找一条边把两个顶点和边去掉，新找的边与原来的不重合。
  天然的关系：match<=OPT
  近似解A=2M<=2OPT

  Match放进顶点，找一个很小的match

- **PTAS**
  不要求知道物品的绝对价值，要求知道物品的相对价值，需要把数据标准化

  假设背包问题有|A-OPT|<k，则背包问题有一个精确解： 背包的value*k+1
  |(k+1)A-(k+1)OPT|<k说明A与OPT一样，因此多项式内近似算法是精确算法

- **近似算法的3种衡量**

1. R（近似比）=A(I)/OPT(I)
   近似比可以是一个常数，也可以是O（logn）
2. R=inf（R<r）
3. 渐进近似比
   有一些近似算法，当求的结论很小的时候没有好的近似算法
   当值提高的时候出在OPT（I）>N的时候出现 好的近似算法

敬爱的李国强老师，

　　您好！

　　我是您19-20第一学期算法导论课程的同学517021910679王浩宇，今早查看教务系统网站，发现算法这门课取得了98分的好成绩，十分感谢李老师！

还有件事儿想麻烦李老师，我打算在1月15日左右申请UCI大学的暑期科研项目，需要填写推荐人信息，目前只需要包含邮箱与电话号码两项，如果UCI方面需要该推荐人的推荐信的话我会写好联系您。不知道您是否愿意做我的refer?

再次感谢李老师这学期的辛勤教导！！！