1. Master定理内容及证明

设a>=1，b>1为常数，f(n)为函数，T(n)为非负整数，且T(n) = aT(n/b) + f(n)，则：

1. 若f(n) = O()，ε>0，那么T(n) = θ()
2. 若f(n) = θ()，则T(n) = θ(logn)
3. 若f(n) = Ω()，ε>0，且对于某个常数c<1和充分大的n有af(n/b)<=cf(n)，则T(n) =θ(f(n))

证明：

T(n) = aT(n/b) + f(n)

设n =

T(n) = aT(n/b) + f(n)

= a[aT(n/) + f(n/b)] + f(n)

= T(n/) + af(n/b) + f(n)

= T(n/) + f(n/) + f(n/) + … + af(n/b) + f(n)

= T(1) +

= + ……………. T(1) =

为所有最小子问题的计算工作量

为迭代过程规约到子问题及综合解的工作量

Case①：T(n) = +

= + O()

……………… 根据case(1)条件放大

= + O()

接下来单独拿出

∵ = =

∴ =

= …………….等比数列求和

∴原式 = + O()

= + O() …………….忽略常数后 =

=θ()

Case②：T(n) = +

= +θ()

= +θ()

= +θ(logn)

=θ(logn)

Case③：T(n) = +

<= +

单独拿出，由条件af(n/b) <= cf(n)，得

<= <= <= … <=

T(n) <= +

= + f(n)

= +θ(f(n))

= θ(f(n))

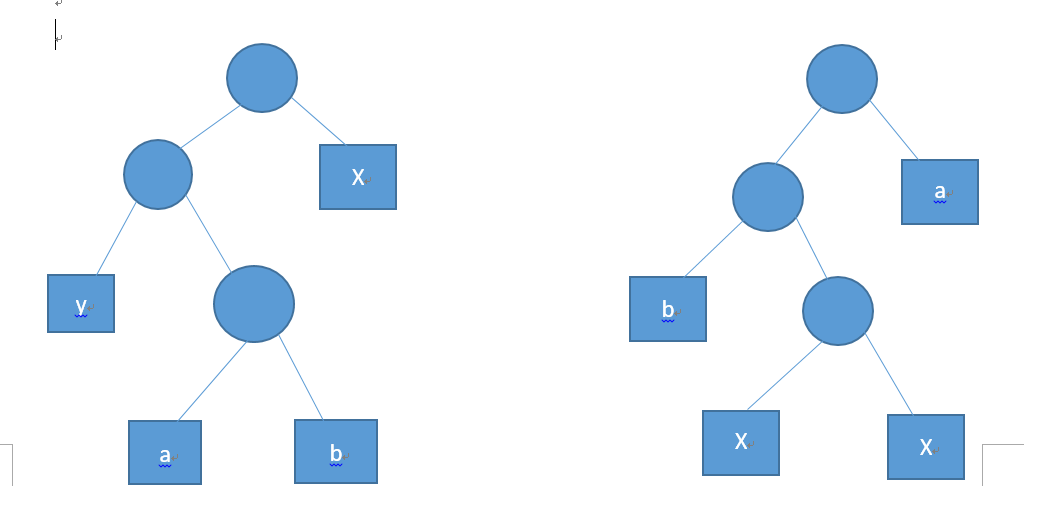
1. 最优前缀码的证明

描述：给定字符集C = {, , … , }和每个字符的频率f()，i = 1, 2, … , n，求关于C的一个最优前缀码(平均传输位数最小)

引理1：C为字符集，任意c∈C，f(c)为频率，x, y∈C，f(x)、f(y)频率最小，那么存在最优前缀码，使得x、y码等长且仅在最后一位不同

f(x) <= f(a)，f(y) <= f(b)

T T’



B(T) – B(T’) = -

= f(x) + 2f(y) + 3f(a) + 3f(b) – f(a) – 2f(b) – 3f(x) – 3f(y)

= -2f(x) – f(y) + 2f(a) + f(b) >= 0

即：变换后权值变小

为i在T中的层数(i到根的距离)

引理2：设T是二元前缀码的二叉树，对于任意x, y∈T，x, y是树叶兄弟，z是x, y的父亲，令T’ = T – {x, y}，且令Z的频率f(z) = f(x) + f(y)，T’是对应二元前缀码C’ = (C – {x, y})∪{z}的二叉树，那么B(T) = B(T’) + f(x) + f(y)

证明：∀c∈C – {x, y}，有 = 🡺 f(c) = f(c)

= = ………………………………. ①

B(T) =

= + f(x) + f(y)

= + f(z) + (f(x) + f(y)) ……..根据①

= B(T’) + f(x) + f(y)

定理：Huffman算法对任意规模为n的字符集C都得到关于C的最优前缀码的二叉树

归纳基础：证明n=2的字符集，Huffman算法得到最优前缀码

归纳步骤：对k的字符集可得到最优，则对k+1也可得到最优

归纳基础：n=2，字符集C = {, }，对任何代码的字符至少都需要1位二进制数，Huffman得到的代码为0或1，是最有前缀码

归纳步骤：假设Huffman对规模为k的字符集得到最优，则对于k+1字符集C = {, , … , }

其中, ∈C是频率最小的两个字符

令C’∈(C – {, })∪{z}，f(z) = f() + f()

根据归纳假设，算法得到关于C’的最优前缀码二叉树T’

把, 作为z的儿子附在T’上，得到树T，则T是关于C = (C – {z})∪{, }的最优前缀码二叉树

如若不然，存在T\*，B(T\*) < B(T)，且由引理1，其树叶兄弟为, ，去掉T\*中的, ，得到T\*’，根据引理2

B(T\*’) = B(T\*) – (f() + f()) < B(T) - (f() + f()) = B(T’)

与T’是一棵关于C’的最优前缀码矛盾

1. Prim算法的证明

Prim(G, E, W)

S = {1}

While v – s 不为空

从v-s种选j使得j到S中顶点的边的权值最小

S = S ∪{j}

正确性证明：归纳法

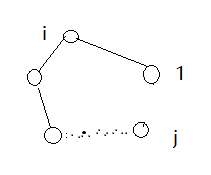
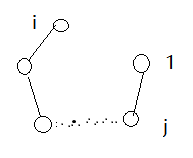
命题：对任意k < n，存在一棵最小生成树包含算法前k步选择的边

归纳基础：k = 1，存在一棵最小生成树T包含边e = {1, i}，其中{1, i}是所有关联1的边中权最小的

归纳步骤：假设算法前k步选择的边构成一棵最小生成树，则前k+1步选择的边构成一棵最小生成树。

归纳基础：存在一棵最小生成树T包含关联节点1的最小权的边e = {1, i}

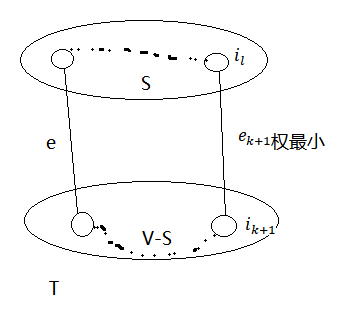
证明：设T位一棵最小生成树，假设T不包含{1, i}，则T∪{{1, i}}包含一条回路，回路中关联1的另一条边{1, j}，用{1, i}替换{1, j}得到树T’，则T’也是生成树，且W(T’)<=W(T)



归纳步骤：假设算法进行了k步，生成树的边为, , … , ，这些边的端点构成集合S，由归纳假设存在G的一棵最小生成树T包含这些边，算法第k+1步选择顶点，则

到S中顶点权最小，设此边 = {, }，若∈T，则算法k+1步显然正确。

假设T不包含，则将加到T中形成一条回路，这条回路有另外一条连接S与V-S中顶点的边e，令T\* = (T – {e}) ∪{}，则T\*是G的一棵生成树，包含、、… 、，且W(T\*) <= W(T)，算法到k+1步仍得到最小生成树。



Kruskal算法的证明

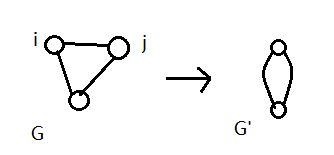
命题：对于任意n，算法对n阶图找到一棵最小生成树

证明思路

归纳基础：证明：n=2，算法正确，G只有一条边，最小生成树就是G

归纳步骤：证明：假设算法对于n阶图是正确的，其中n>1，则对任意n+1阶图算法也得到一棵最小生成树

链接操作：任给n+1个顶点的图G，G中最小权边e={i, j}，从G中短接i和j，得到G’



归纳步骤：对于任意n+1阶图G短接最短边e，得到n阶图G’，根据归纳假设算法得到G’的最小生成树T’，将被短接的边e“拉伸”回到原来长度，得到树T，证明T是G的最小生成树

T=T’∪{e}是关于G的最小生成树，否则假设存在G的含边e的最小生成树，W(T\*)<W(T)(如果e不属于T\*，在T\*中加边e，形成回路，去掉回路中任意别的边所得到的生成树权值仍旧最小)，在T\*短接e得到G’的生成树T\*-{e}

W(T\*-{e}) = W(T\*) – W(e) < W(T) – W(e) = W(T’)

与T’的最优性矛盾

单源最短路径

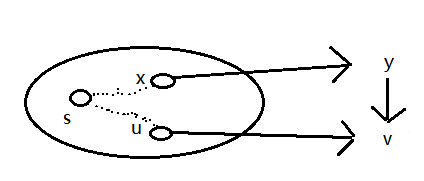
命题：算法进行到第k步时，对于S中每个节点i，dist[i] = short[i](局部最短等于全局最短)。

归纳基础：k = 1，S = {s}，dist[s] = short[s] = 0

归纳步骤：证明假设命题对k为真，则对k+1命题也为真

归纳步骤：选择顶点v(边<u, v>)，需证明dist[v] = short[v]

若存在另一条s-v路径L，最后一次出S的顶点为x，经过V-S的第一个顶点y，再由y经过一段再V-S中的路径到达v



在k+1步算法选择顶点v，而非y，所以dist[v]<=dist[y]，令y到v的路径长度为d(y, v)，

则dist[y] + d(y, v) <= L，于是dist[v] <= L，即dist[v] = short[v]