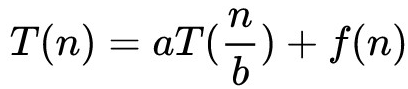
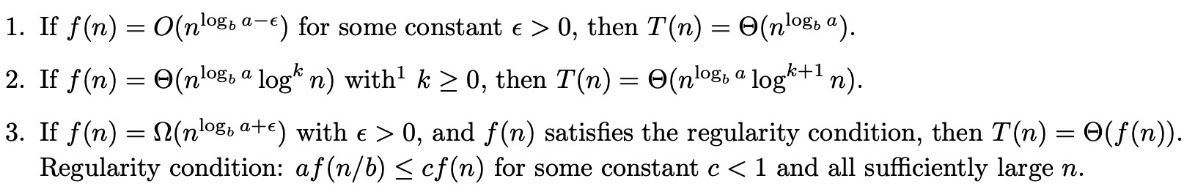
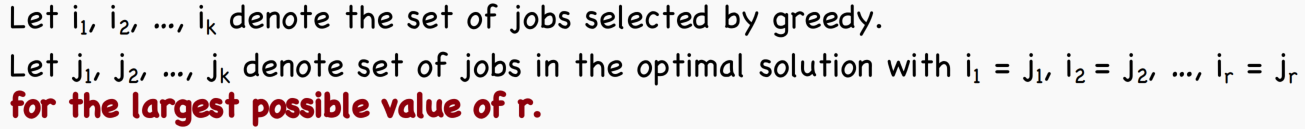
**Poly-time**: there exist constants c>0 and d>0 such that on every input of size N, its running time is bounded by cN­­­d steps.

**Upper bounds**: T(n) is O(f(n)) if there exist constants c>0 and n0>=0 such that for all n>=n0 we have T(n) <= c·f(n)



**贪心证明**：



**Exist constants c1, c2 , n0 , such that c1f(n) <= T(n) <= c2f(n) for all n >=n0**



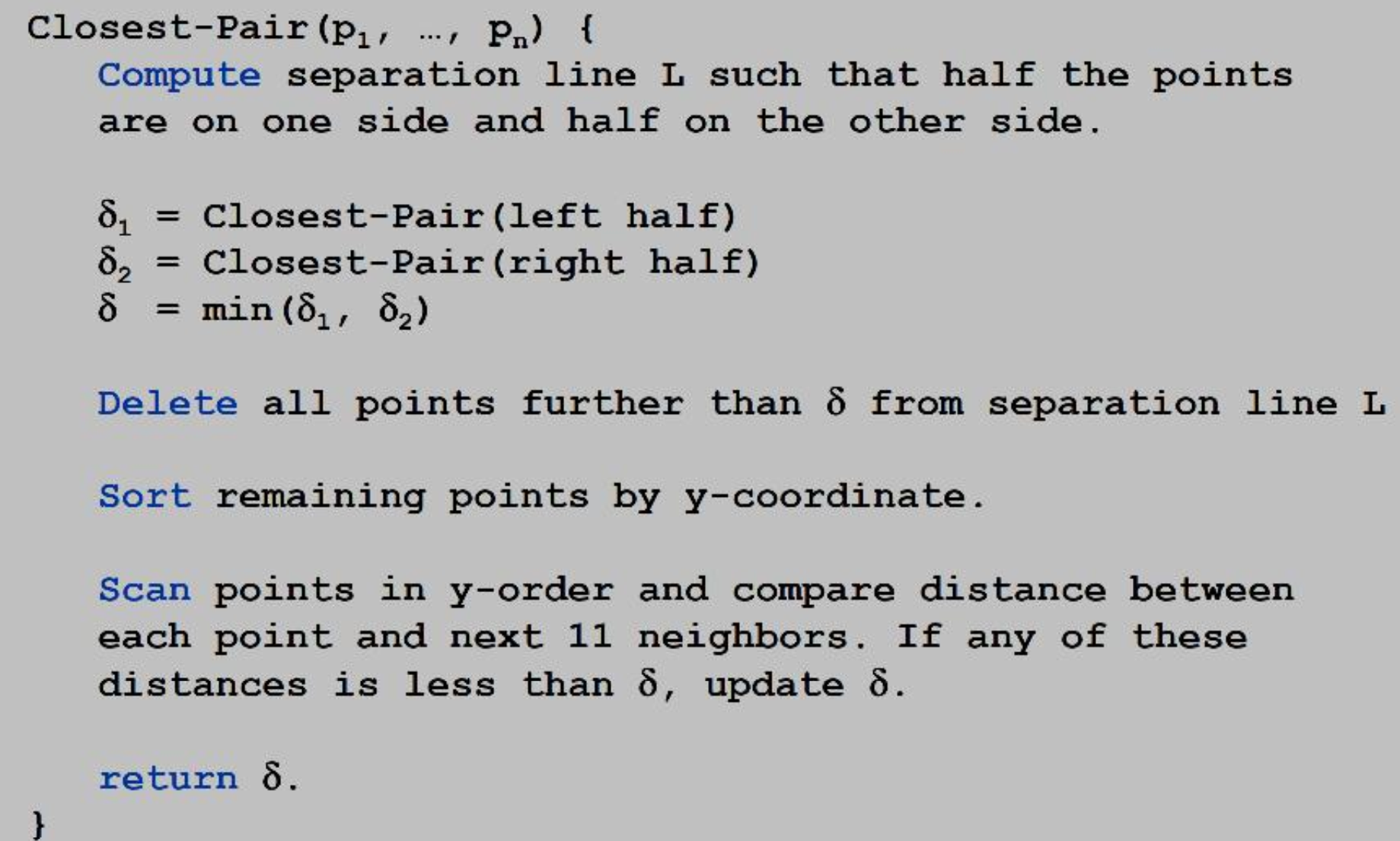
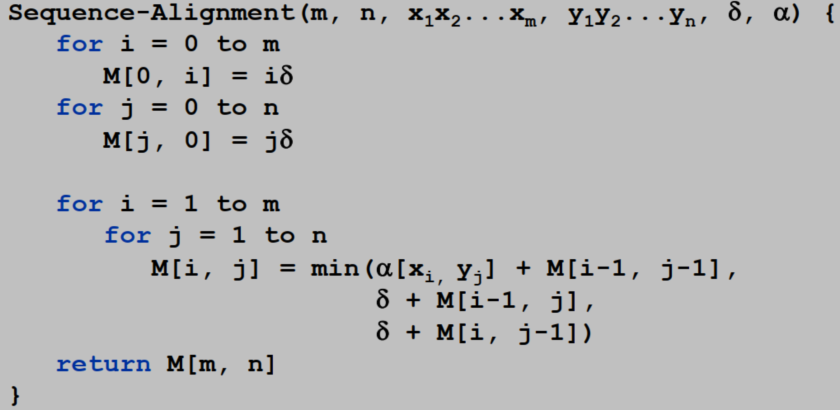
**Cut lemma**: Let S be any subset of nodes, and let e be the min cost edge with one endpoint in S. Then every MST must contain e

Prim: 逐个加点 Kruskal：逐条加边

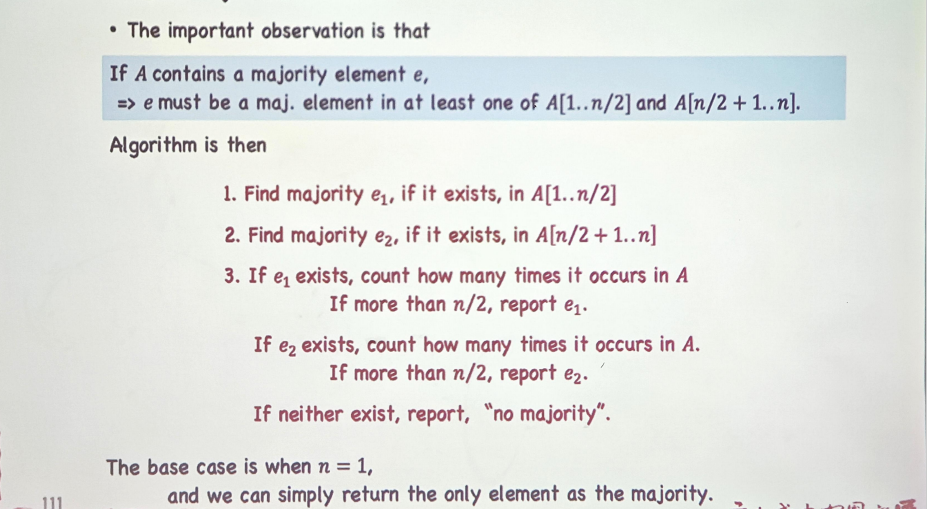
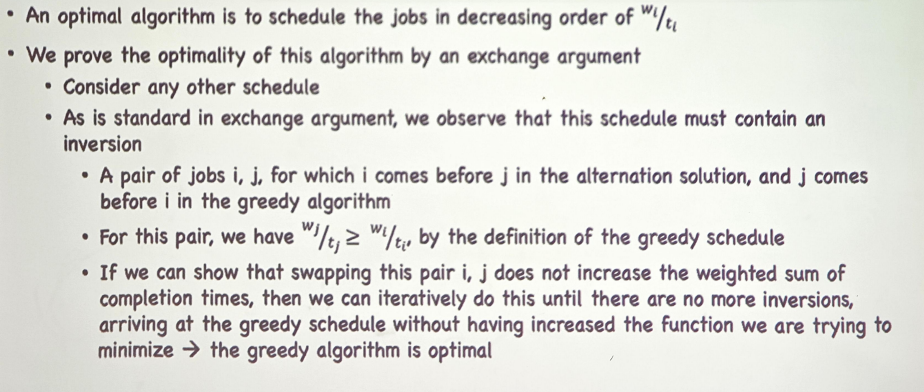
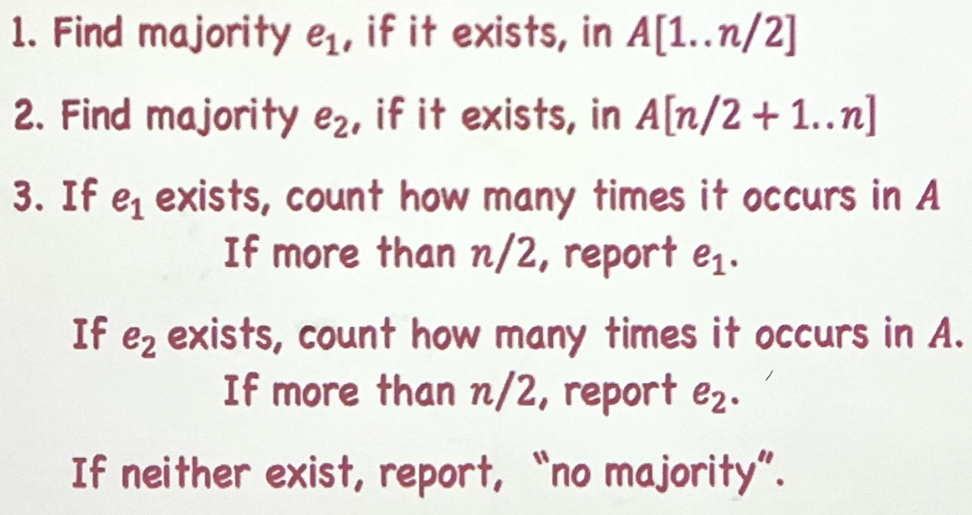
**Proof of Kruskal**: 1.prove Kruskal produce a tree(acyclic and connected graph); 2.prove every edge chosen is in MST(Cut lemma); 3. Prove tree T is a MSL;

**Reduced schedule claim**：any unreduced schedule can transform into reduced schedule with no more eviction

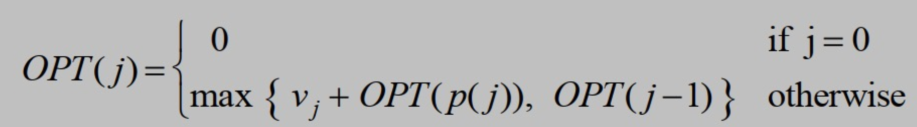
**Closest-pair**: T(n) = O(nlog2n)，可以优化至O(nlogn)

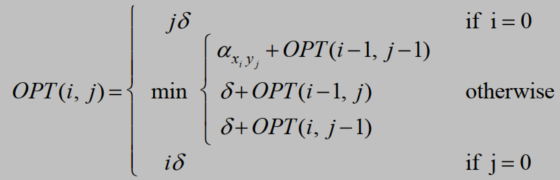
**Divide-and-conquer**: Break up a problem into a few sub-problems, solve each sub-problem independently and recursively, and combine solution to sub-problems to form solution to original problem.



**Dynamic programming**: Break up a problem into a series of overlapping sub-problems, and build up solutions to larger and larger sub-problems.

**Weighted interval scheduling**: 

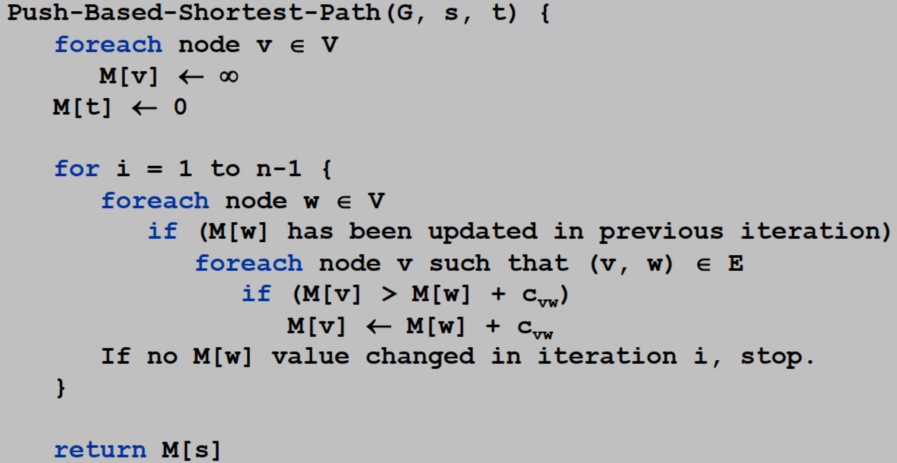
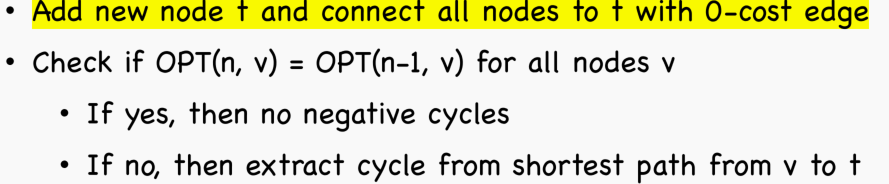
**Sequence Alignment**: OPT(i,j) = min cost of aligning string x1...xi and y1...yj. time:O(mn) space:O(mn) 🡪 space:O(m+n)



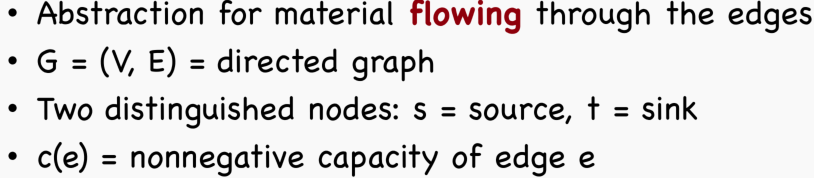
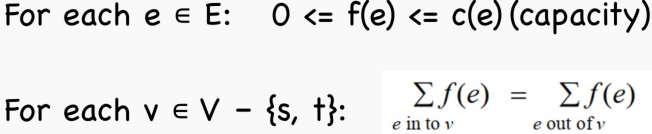
**Shortest Paths**: OPT(i,v) = length of shortest v-t path P using at most i edges. 🡪 Bellman-ford algorithm

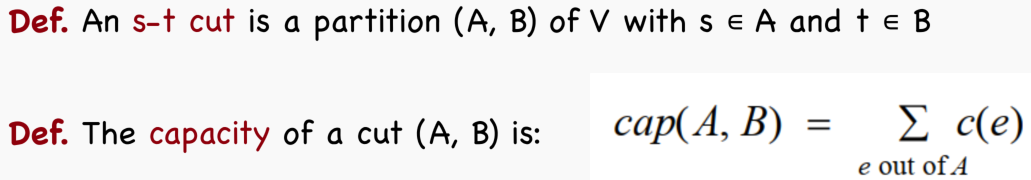
**Detecting Negative Cycles**: If OPT(n,v) = OPT(n-1,v) for all v, then there is no negative cycle with a path to t.

We can use Bellman-ford to detect negative cost cycle in O(mn) time.

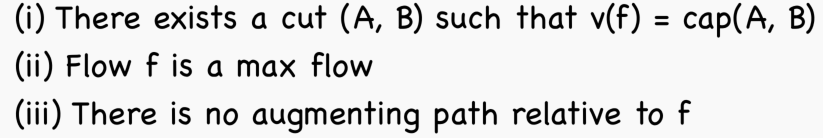
**Flow network / s-t flow**



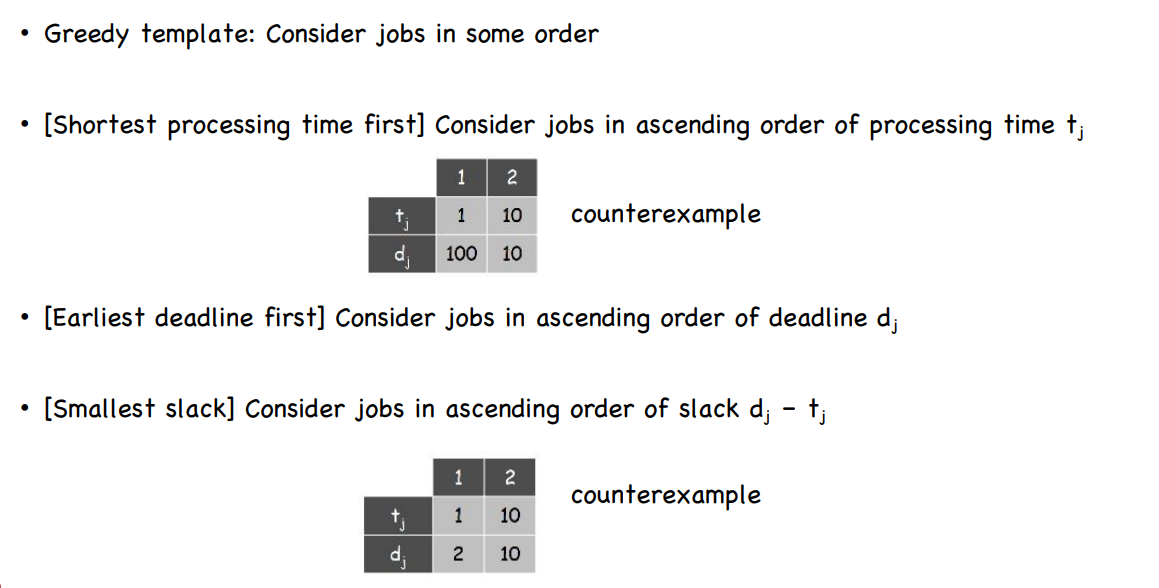
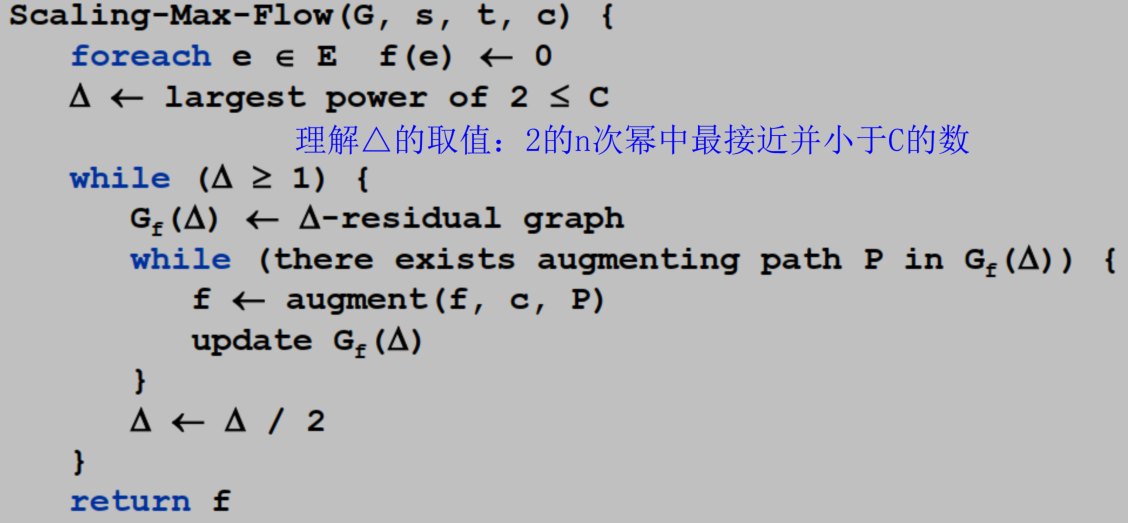
**最小割与网络流的弱对偶**：v(f) ≤ cap(A, B)

**最大流最小割定理**：the value of the max flow is equal to the value of the min cut. 以下是证明：



(i)🡪(ii): weak duality (ii)🡪(iii): augmenting path theorem (iii)🡪(i): residual gragh

通过寻找最优路径来加速最大流求解 time: 从O(mnC) 降低至**O(m2logC)**



**Lemma 1**: let f be the flow at the end of a △-scaling phase, then the value of the maximum flow is at most v(f)+m△

**Lemma 2**: there are at most 2m augmentation per scaling phase

