Graph Algorithm III

Instructor: Shizhe Zhou

Course Code:00125401

Dijkstra shortest path

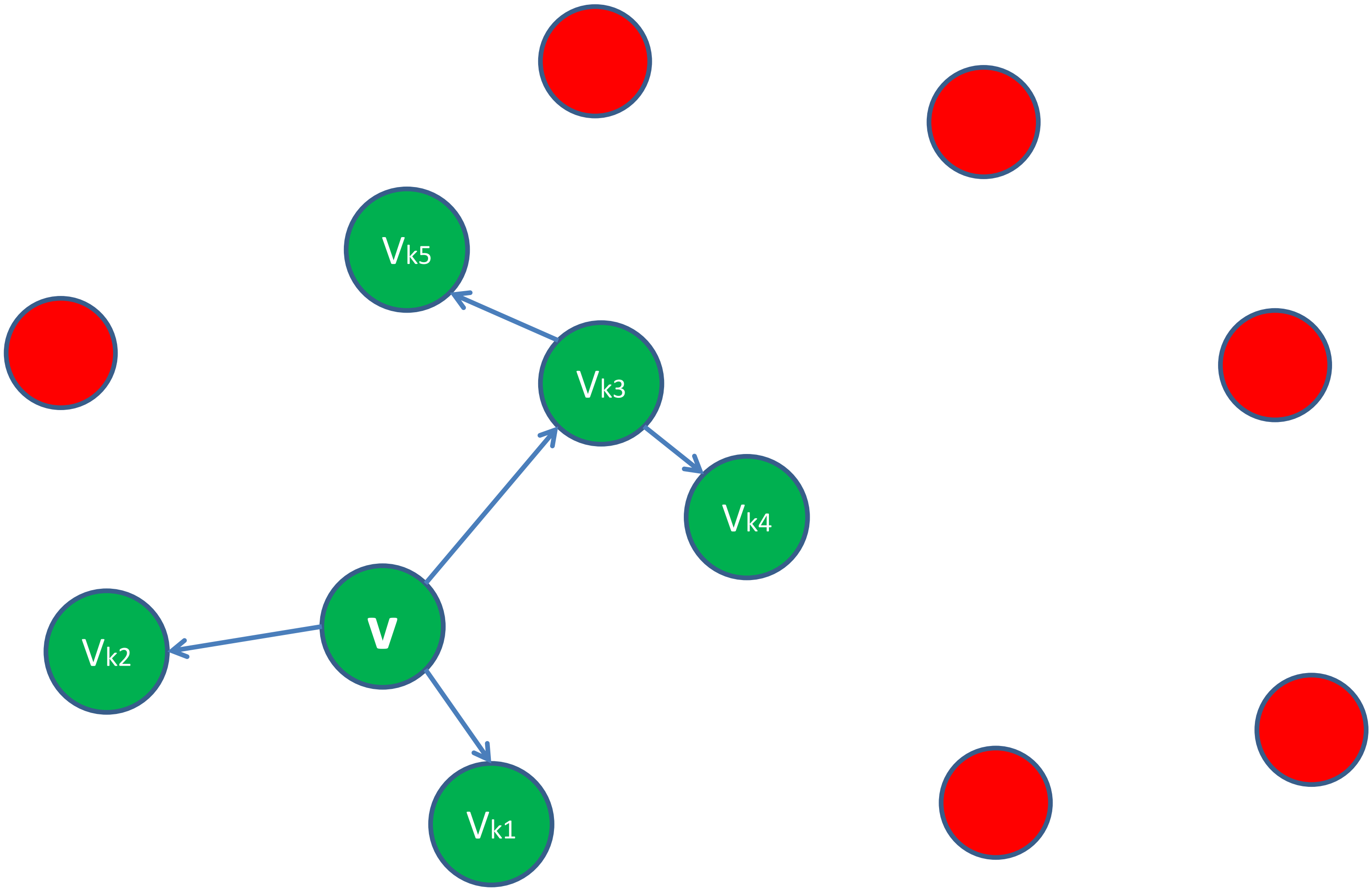
- **策略： 找下一个最靠近的顶点.**

- 方法： 下一条最短路径只可能是从V_k所定义的某条路径**末端增加一条边**所得.

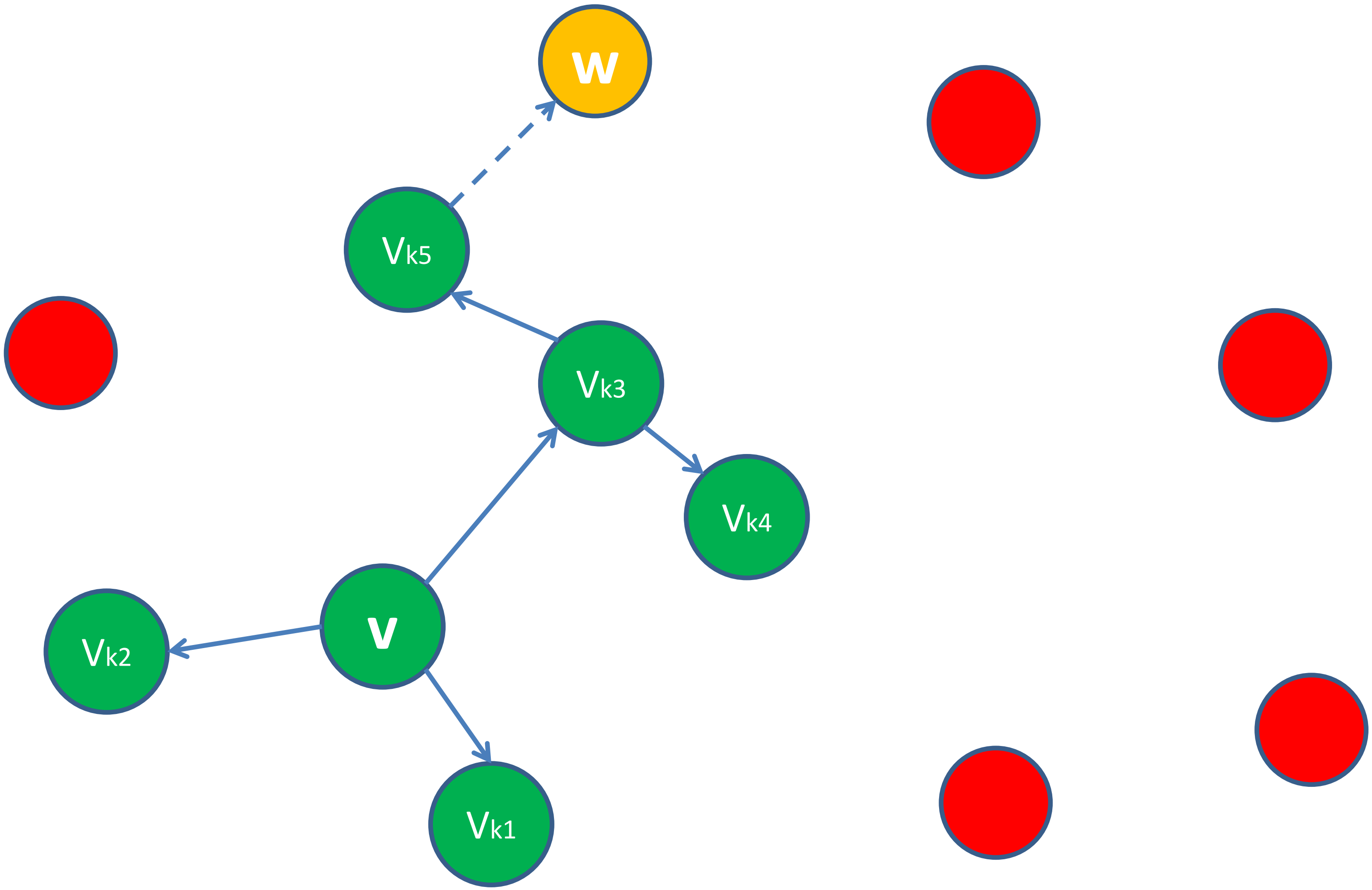
- 无需调整：对于含有cycle的有向图, 没有topological order， 如何保证新发现的顶点不会减小已发现顶点的最短路径？

不适用于带有负权边的图！

- 方法： 下一条最短路径只可能是从V_k所定义的某条路径**末端增加一条边**所得.

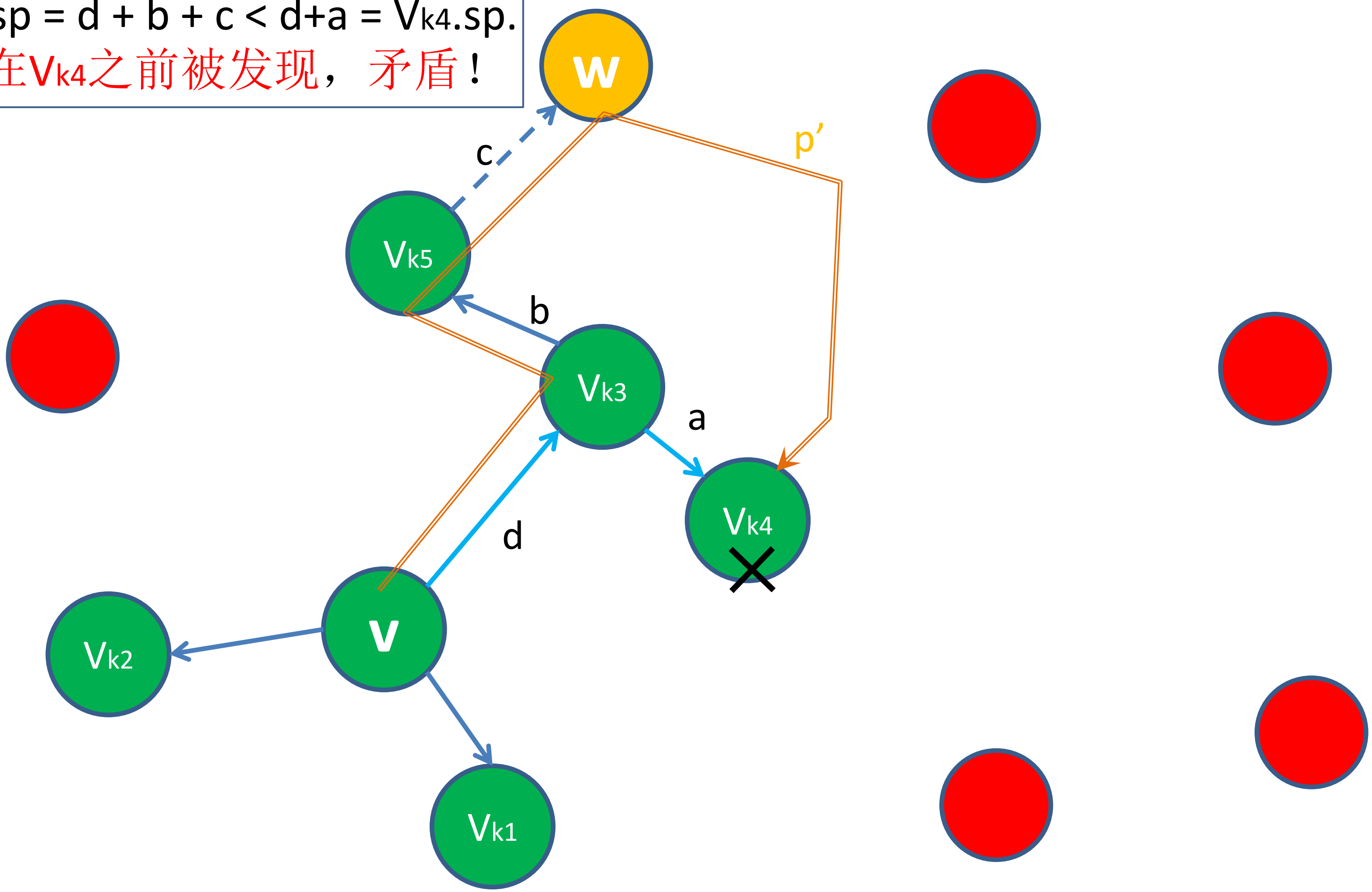


- 方法：下一条最短路径只可能是从V_k所定义的某条路径**末端增加一条边**所得.



- 无需调整：如何保证新发现的顶点不会减小已发现顶点的最短路径？

若不然，有d+a>d+b+c+p’, →a>b+c
从而使得W.sp = d + b + c < d+a = Vk4.sp.
那么W应该在Vk4之前被发现，矛盾！

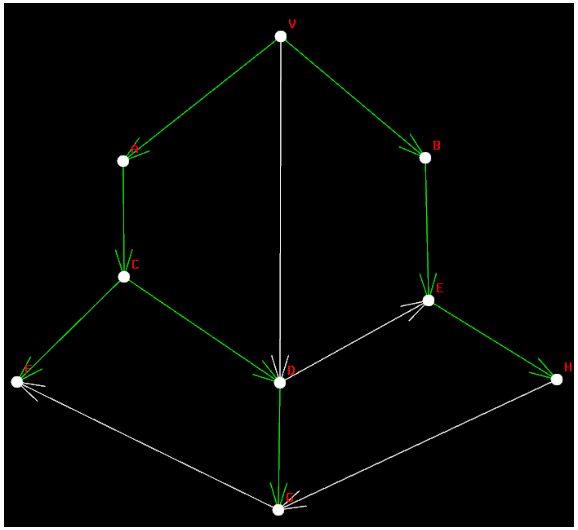


Single source shortest path code

Code:

http://staff.ustc.edu.cn/~szhou/course/algorithmFundamentals/graphalgo.zip

Dijkstra Tree.



适定性

- Dijkstra

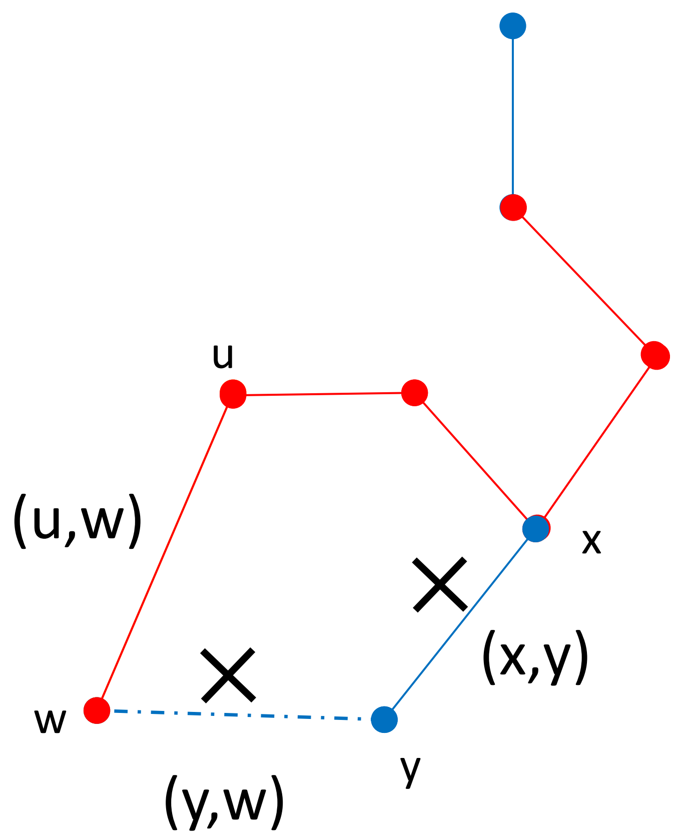
--适用于有向图与无向图, 不含任何负权边.

- Bellman-Ford

--适用于带有负权的边，但是不存在负权边组成的回路的图！

Min-Cost Spanning Tree

- Min edge(u, w) must belong to MCST

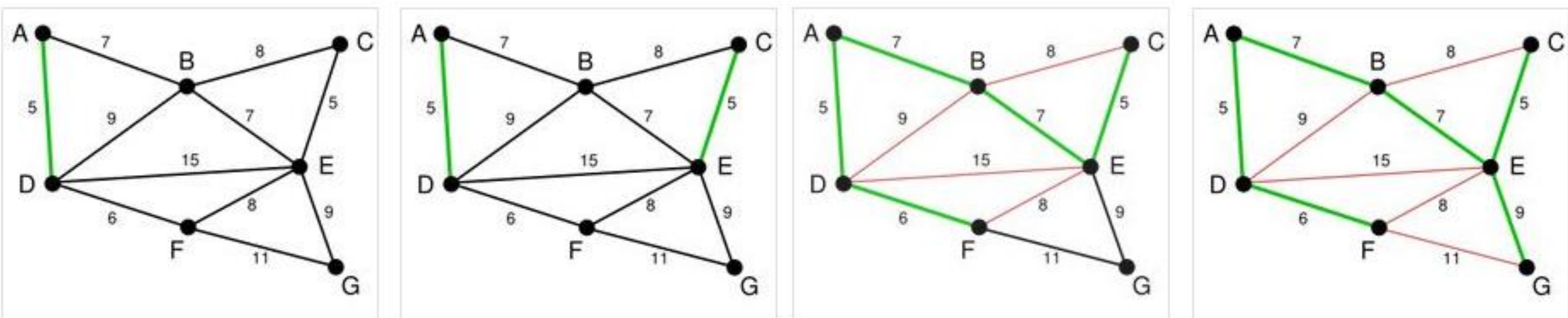


a. T如何演化成MCST?通过加入最小权的外接边(u,w).

b. 如何保证(u,w)必然属于最终的MCST?
若不然:因T是MST的一部分. u和w必须使用一条路径相连通,该路径通过 T中已存在的另外一个顶点x.
因为(u,w)是连接T与T之外顶点的边中权最小的,
故: (x,y) >(u,w)。
在形成的闭圈中，可将(x,y)替换为(u,w)，同样达到了新顶点w,，且不影响闭圈中其余顶点的连通性，得到的MCST优于原来不包含(u,w)的伪MCST.

Kruskal

- kruskal算法总共选择n- 1条边，（共n条边）所使用的贪婪准则是：从剩下的边中选择一条不会产生环路的具有最小耗费的边加入已选择的边的集合中。注意到所选取的边若产生环路则不可能形成一棵生成树。kruskal算法分e 步，其中e 是网络中边的数目。按耗费递增的顺序来考虑这e 条边，每次考虑一条边。当考虑某条边时，若将其加入到已选边的集合中会出现环路，则将其抛弃，否则，将它选入。



Prim and Kruskal

- Kruskal算法work for unconnected graph
- 对于稠密图性能较高的算法是Prim.