

最短路径Dijkstra

1. 算法场景描述

如今出行已经不需要再为找不着路而担心了，车上有车载导航，手机中有导航App。只需要确定起点和终点，导航会自动规划出可行的距离最短的道路。这是最短路径在人们实际生活中最典型的应用。

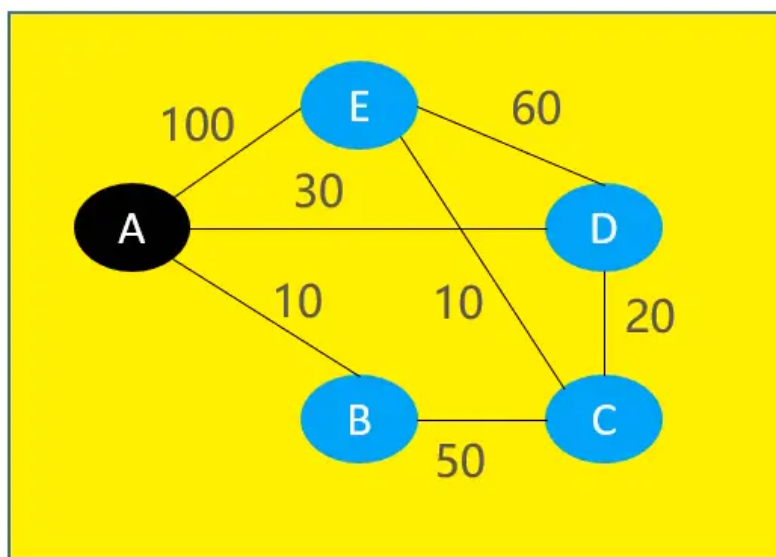
2. 算法说明

- dijkstra算法是求单源点到其他结点的最短路径的方法，所以其他任何节点，都是依赖于源点出发的。
- 这样，肯定是先找源点能到节点的最短路径，在这个基础上，通过确定的最短点再次发散到其他结点，继续找最小，以此类推，把所有结点都找完为止。
- 找的过程，要标记已经找到的结点，这样避免环路出现。

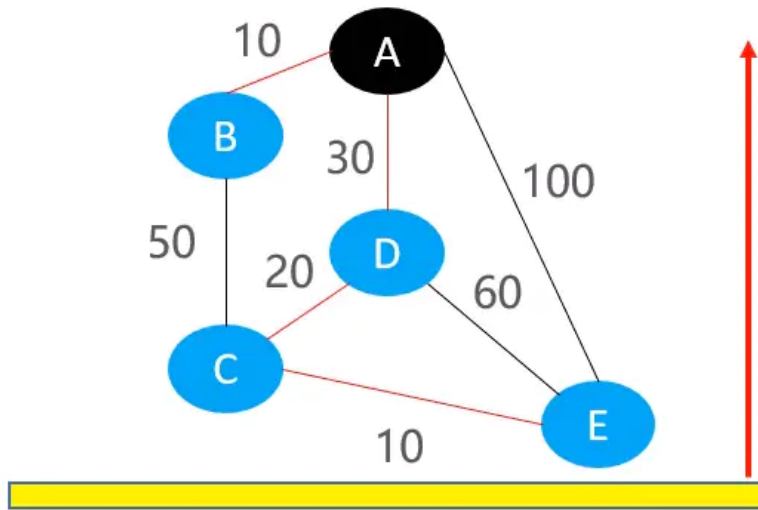
Dijkstra等价思考

首先，Dijkstra的原理其实和生活中的一些自然现象是完全一样的。什么自然现象呢？接下来就一起跟着下面的描述，来想想一件事情。如果想清楚的话，对于理解Dijkstra算法会大有帮助。

- 1.把图中的每一个顶点都想象成一块小石头。
- 2.每一条边想象成是一条绳子，每一条绳子都连接着2块小石头，边的权值就是绳子的长度。
- 3.将小石头和绳子平放在一张桌子上，（下图是一张俯视图，图中黄颜色的是桌子）



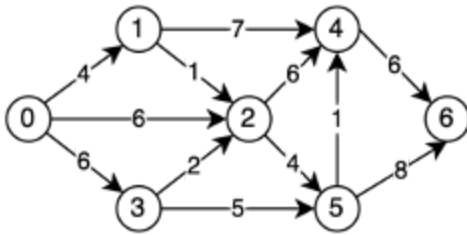
- 4.接下来想象以下，用手拽着小石头A，慢慢地向上提起来，远离桌面，如下图所示的侧视图



5.离开桌面的顺序取决于其余顶点距离顶点A的最短绳子长度，也就是最短路径。所以，最终你会发现，B,D,C,E会依次离开桌面，当所有石头都离开桌面时，有的绳子会蹦直，而有的绳子是松的。

6.最后绷直的绳子就是A到其他小石头的最短路径。

关键信息：后离开桌面的小石头，都是被先离开桌面的小石头拉起来的。



引入3个辅助数组dist[], path[], check[]

dist描述的是源点到其他结点的距离，初始化时候只关心源点到他的邻接点的距离；

path描述的是到该节点前的经过哪个顶点到的

check描述的是该节点是否已经被访问