# K短路

顾名思义，就是求图上从起点到终点的第k短路。

算法策略有两种，A\*搜索和可持久化堆，k较小时候二者差不多，k较大搜索会变得很慢，可持久化堆是较好解法。

## A\*

A\*搜索概念和模型参考另一个文章。

A\*是广义广搜，估价函数f(x)=g(x)+h(x);g(x)作为实际走的路径就定义为起点到点x的最短距离，h(x)作为预计到达目标状态的值，定义为x到终点最短距离，可以通过建反向图跑最短路的得到。

建立优先队列，有2个属性(key,e) ,key是f(x)值，把key较小的放在堆顶，e代表下一个点编号，tcnt是代表了终点T出队次数。之后开搜索，实际上就是进行了若干次无松弛的spfa或者说用优先队列的bfs，每次出队一个点，要把所有与它相连的点入队列，v.key=f(v)=u.key-dist[u]+w(u,v)+dist[v]

为什么是这样，u.key-dist[u]相当于g(x)，是当前走过的路径长加上w(u,v)后，是v点已经走过路径长，再加上dist[v]自然就是f(v)

## 可持久化可并堆