INT1 0 2 W 5 _ Dijkst ra

定义： 基于贪心算法的单源最短路径算法，即基于某一点出发，要求最后得到该点到图内其他任意点的最小距离。

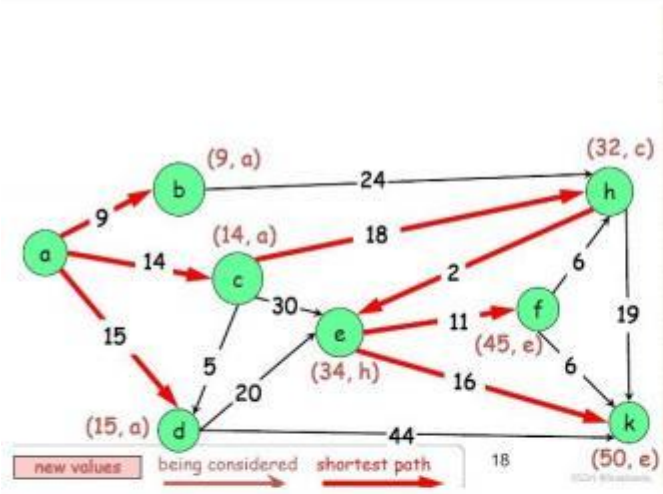
适用范围： 有向图与无向图，通常用来解决比如导航之类的问题。

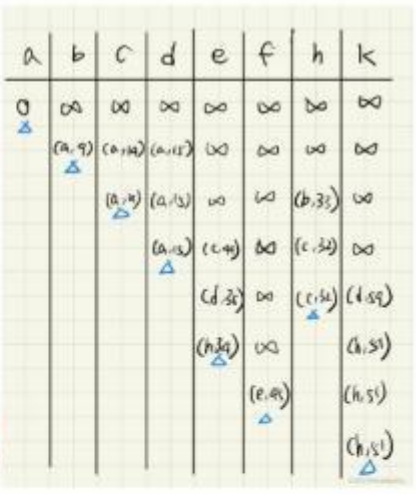
实现方式： 区别于 prim 算法，它并不使用递归实现而是使用堆和迭代来进行。

实现流程： 对于该算法而言，他主要是分为两个步骤，

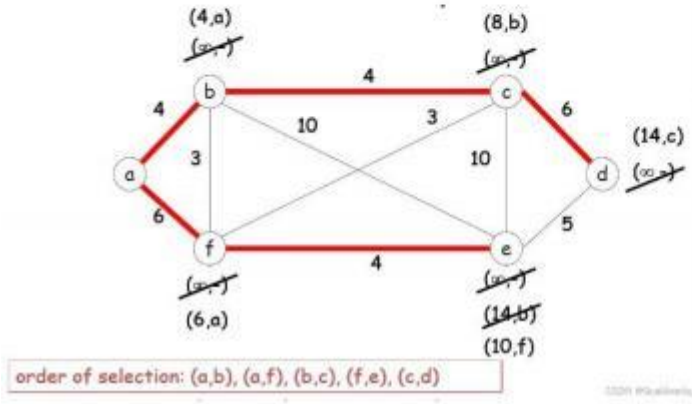
第一步将所有的非起点距离设置为无穷大，然后再这基础上标记距离 A 最近的点（未标记状态），更新距离，收入最佳路径中，然后计算上一节点 A 到刚标记节点 B 的距离与节点 A 到出发点的距离的和，若小于 B 到原点的距离则锁定 A 点，更新到 B 的距离。

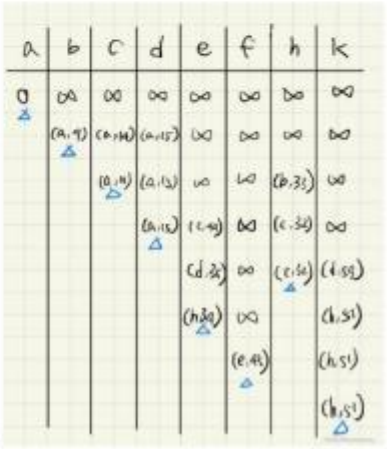
图示表达：





例2





代码实现：

Pseudo code

```
// Given a graph G=(V,E) and a source vertex s
for every vertex v in the graph do
    set d(v) = ∞ and p(v) = null
set d(s) = 0 and V_T = ∅          (this should be ∅)
while V - V_T ≠ ∅ do      // there is still some vertex left
begin
    choose the vertex u in V - V_T with minimum d(u)
    set V_T = V_T ∪ {u}
    for every vertex v in V - V_T that is a neighbour of u do
        if d(u) + w(u,v) < d(v) then      // a shorter path is found
            set d(v) = d(u) + w(u,v) and p(v) = u
end
```

代码的距离流程我认为是这样的，首先我们给出一个图像G，然后让点位V，边为E，对于所有的图内顶点V 我们设定将每个顶点的距离初始化为无穷大（(∞)），前驱顶点初始化为null，在这之后我们把出发点的距离初始化为0 并且令存储已访问并更新节点的集合V 为空。然后当任然存在没有记录更新的点的时候，我们从中选择对于距离最短的点，将他加入到已记录集合 VT 中，然后遍历所有和她相邻的节点，检查当前通过顶点 (u) 的路径是否比先前已知的最短路径更短，如果更短那么就更新相邻点的最短距离和前置顶点。

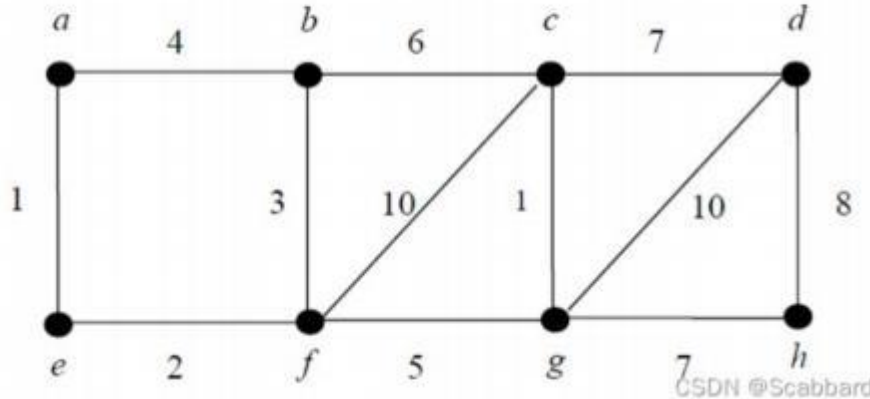
各变量代表的定义：

D（v）=距离，p（v）=前一个顶点，s=原点，v=点的所有集合，vt= 已标记点

。 例题：

Question 3

Consider the following graph G. The label of an edge is the cost of the edge.



3. Referring to the same graph above, find the shortest paths from the vertex *a* to *all* other vertices in the graph G using *Dijkstra's* algorithm. Show the changes of the priority queue step by step and give the order in which edges are selected. **(5 marks)**

N.B. There may be more than one solution. You only need to give one of the solutions.

Order Selected	a(0,-)	b(-,∞)	c(-,∞)	d(-,∞)	e(-,∞)	f(-,∞)	g(-,∞)	h(-,∞)
a(0,-)		b(a,4)	c(-,∞)	d(-,∞)	e(a,1)	f(-,∞)	g(-,∞)	h(-,∞)
e(a,1)		b(a,4)	c(-,∞)	d(-,∞)		f(e,3)	g(-,∞)	h(-,∞)
f(e,3)		b(a,4)	c(f,13)	d(-,∞)			g(f,8)	h(-,∞)
b(a,4)			c(b,10)	d(-,∞)			g(f,8)	h(-,∞)
g(f,8)			c(g,9)	d(g,18)				h(g,15)
c(g,9)				d(c,16)				h(g,15)
d(c,16)								h(g,15)
h(g,15)								

对于这个题我们可以通过上表的方式来进行思考，每一列代表在这一次循环中做了什么。