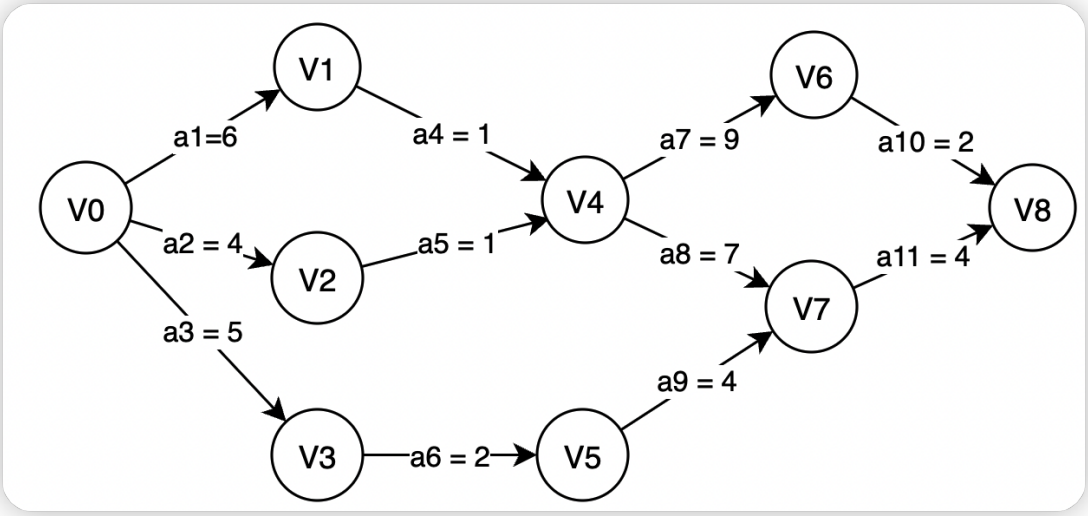
关键路径

一、基本概念

1. 什么是AOE网

AOE网（Activity On Edge）即边表示活动的网，是与AOV网（顶点表示活动）相对应的一个概念。而拓扑排序恰恰就是在AOV网上进行的，这是拓扑排序与关键路径最直观的联系。AOE网是一个带权的有向无环图，其中顶点表示事件（Event），弧表示活动，权表示活动持续的时间。下面的就是一个 AOE网：



其中V0，V1，V2.....V8表示事件，a1.....a11表示活动，活动的取值表示完成该活动所需要的 时间，如a1 = 6表示完成活动a1所需要的时间为6天。此外，每一事件Vi表示在它之前的活动已经完成， 在它之后的活动可以开始，如V4表示活动a4和a5已经完成，活动a7和a8可以开始了。

2. AOE网的源点和汇点

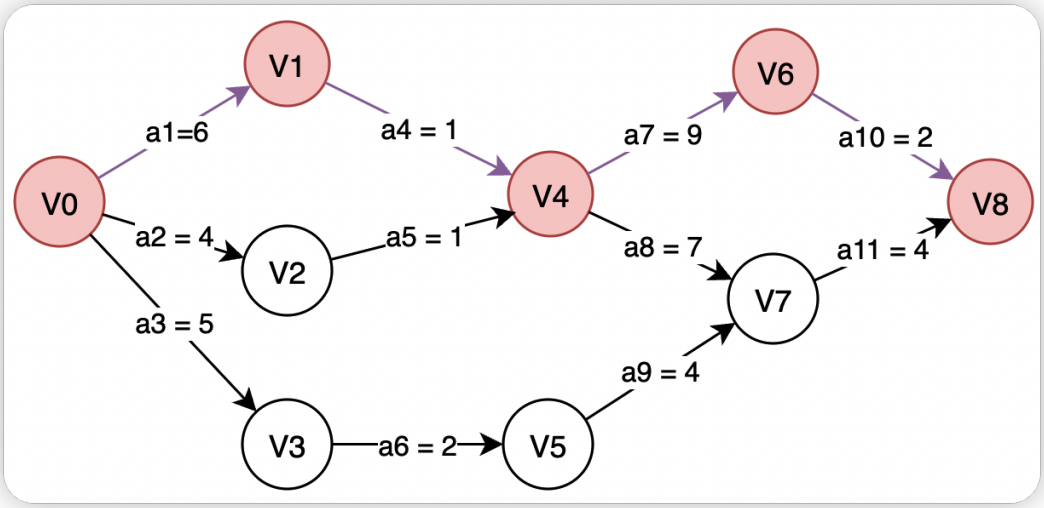
由于一个工程中只有一个开始点和一个完成点，故将AOE网中入度为零的点称为源点，将出度为 零的点称为汇点。

打个比方，我们现在有一个工程，就是将大象装进冰箱，那么源点就相当于我们现在接到这样一 个任务，而汇点则表示我们完成了这个任务。那么我们之前所讲的打开冰箱门，将大象装进去，关上冰 箱门就属于活动本身（即a1....a11所表示的信息），打开冰箱门所需要的时间就是活动所需要的时间， 而完成某一个活动所到达的顶点就表示一个事件（冰箱门打开）。上图中的顶点V0表示源点， V8表示汇点。

3. 什么是关键路径

唐僧师徒从长安出发去西天取经，佛祖规定只有四人一起到达西天方能取得真经。假如师徒四人分别从长安出发，走不同的路去西天：孙悟空一个筋斗云十万八千里，一盏茶的功夫就到了；八戒和沙和尚稍慢点也就一天左右的时间；而唐僧最慢需要14年左右。徒弟到达后是要等着师傅的。那么用时最长的唐僧所走的路，就是取经任务中的关键路径。其他人走的路径属于非关键路径。

由于AOE网中的有些活动是可以并行进行的（如活动a1、a2和a3就是可以并行进行的），所以完成工程的最短时间是从源点到汇点的最长路径的长度。路径长度最长的路径就叫做关键路径（Critical Path）。如下图中红色顶点和有向边构成的就是一条关键路径，关键路径的长度就是完成活动 a1、a4和a9、a10所需要的时间总和，即为 6+1+9+2 = 18。



4. 什么是ETV

ETV（Earliest Time Of Vertex）：事件最早发生时间，就是顶点的最早发生时间；

事件V1的最早发生时间表示从源点V0出发到达顶点V1经过的路径上的权值之和，从源点V0出发到达顶点V1只经过了权值为6的边，则V1的最早发生时间为6，表示在活动a1完成之后，事件V1才可以 开始；

同理，事件V5要发生（即最早发生）需要活动a3和活动a6完成之后才可以，故事件V5的最早发生时间为 5 + 2 = 7。其他顶点（事件）的最早发生时间同理可的。需要说明，事件的最早发生时间一定是从源点到该顶点进行计算的。

5. 什么是LTV

LTV（Latest Time Of Vertex）：事件最晚发生时间，就是每个顶点对应的事件最晚需要开始的时间，如果超出此时间将会延误整个工期。

前面在谈关键路径的概念时给出了一条上图中的关键路径，该关键路径（ V0，V1，V4 ，V6，V8）的长度为18，为什么要提这个长度呢，因为要计算某一个事件的最晚发生时间，我们需要从汇点V8进行倒推。计算顶点V1的最晚发生时间为例，已知关键路径的长度为18，事件V1到汇点V8所需要的时间

为 1 + 9 + 2 = 12，则事件V1的最晚发生时间为18–12 = 6，这时候我们发现，这和事件V2的最早发生时间不是一样吗？的确如此，对于关键路径上的顶点都满足最早发生时间 etv 等于 最晚发生时间 ltv 的情况，这也是我们识别关键活动的关键所在。

再来计算一下事件V5的最晚发生时间，事件V5到汇点V8所需 要的时间为 4 + 4 = 8，则事件V5的最晚发生时间为 18 – 8 = 10；相当于说活动a6完成之后，大可以休息 3天，再去完成活动a9也不会影响整个工期。

6. 什么是ETE

ETE(Earliest Time Of Edge)：活动的最早开工时间，就是弧的最早发生时间，就是事件的最早发生时间。

活动a4要最早开工时间为事件V1的最早发生时间 6；同理，活动a9的最早开工时间为事件v5的最早发生时间 7。显然活动的最早开工时间就是活动发生前的事件的最早开始时间。

7. 什么是LTE

LTE(Lastest Time of Edge)：活动的最晚开工时间，就是不推迟工期的最晚开工时间。

活动的最晚开工时间则是基于事件的最晚发生时间。比如活动a4的最晚开工时间为事件V4的最晚发生时间减去完成活动a4所需时间，即 7 – 1 = 6；活动 a9的最晚开工时间为事件V7的最晚发生时间减去完成活动a9所需时间，即 14 – 4 = 10；

从上面也就可以看出 只要知道了每一个事件（顶点）的ETV 和 LTV，就可以推断出对应的 ETE 和 LTE . 此外还需要注意，关键路径是活动的集合，而不是事件的集合，所以当我们求得 ETV 和 LTV 之后，还需要计算 ETE 和 LTE 。

二、关键路径算法

求关键路径的过程事实上最重要的就是上面提到的四个概念，ETV、LTV、ETE 和 LTE，求得了ETE与LTE之后只需要判断两者是否相等，如果相等则为关键路径中的一条边，则输出。

A. 拓扑排序获得每一个事件的最早发生时间ETV

- 拓扑排序获得每一个事件的最早发生时间ETV

  1. 初始化将ETV当中的数据设置为0
  2. 将入度为0的顶点V0入栈　　（V0） ---> V0
  3. 顶点V0出栈，遍历V0的邻接顶点

     a. 将邻接顶点的入度减一，如果为0，则进栈

     b. 进栈时，判断ETV[0] + w(v0,v1)是否大于ETV[1]，为真就更新ETV[1]

     （V1, V2, V3） ---> V3
  4. 弹栈V3，遍历顶点V3的邻接点V5，将V5的入度减一，更新ETV[5] = ETV[3] + W(3,5)

     a. (V1, V2, V5) ---> V5
  5. 弹栈V5，遍历顶点V5的邻接点V7，将V7的入度减一，并更新ETV[7]

     a. (V1, V2) ---> V2
  6. 弹栈V2，遍历邻接顶点V4，更新ETV[4]

     a. (V1) ---> V1
  7. 弹栈V1，遍历邻接顶点V4，更新ETV[4]

     a. (V4)
  8. 弹栈V4，遍历得到V6,V7,更新ETV[6],ETV[7],入栈入度为0

     a.(V6, V7) ---> V7

  ... --> V6 --> V8

- 推导LTV

  上一步的入栈后要进行出栈，其出的顺序就是倒序的顺序

  V8 V6 V7 V4 V1 V2 V5 V3 V0

  1. V8取出，V8没有出度，什么都不做
  2. V6取出，判断LTV[6] – w(6, 8) 小于