关键路径的java实现Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E5_85_B3_E9_94_AE_E8_B7_AF_E5_c104_645069.htm /* * @title：关键路径 * @input: 有向带权图，图以邻接表形式表示，头结点只存储该顶点的度，后继结点存储顶点及权值 * @output: 所有可能关键路径的并集path，path[i][0]及path[i][1]代表边的顶点，path[i][2]代表权值 */ import java.util.*. public class CriticalPathTest { public static void main(String[] args) { int[][] graph={{0, 1,6, 2,4, 3,5,},{1, 4,1,},{1, 4,1},{1, 5,2,}, {2, 6,9, 7,7,},{1, 7,4,},{1, 8,2,},{2, 8,4,},{2,},}. int[][] path. CriticalPath criticalPath=new CriticalPath(). criticalPath.input(graph). path=criticalPath.getPath(). for(int i=0. i System.out.println("边：" path[i][0] "-" path[i][1] " 权：" path[i][2]). } } } class CriticalPath { private int[][] graph. private int[][] path. int len. void input(int[][] graph) { this.graph=graph. path=new int[graph.length-1][]. len=0. calculate(). } void calculate() { int[] ve=new int[graph.length]. //事件的最发生时间 Stack stack1=new Stack(). Stack stack2=new Stack(). int i,j,v. for(int t : ve) t=0. stack1.push(0). while(stack1.empty()!=true){ v=(Integer)stack1.pop(). for(i=1. i j=graph[v][i]. if((--graph[j][0])==0){ stack1.push(j). }