

关键路径 critical Path

1. 概念: ~~AOE~~

AOE网(边活动网): 带权有向图, 顶点表示事件, 有向边表示活动, 边的权值表示开销。

关键路径: 从开始顶点到完成顶点的最长路径。

关键活动: 关键路径上的活动。

事件 v_i 可能的最早发生时间 ~~最早~~ — $ee(v_i)$;

v_i 允许的最迟发生时间 — $le(v_i)$;

活动 $a_k = \langle v_i, v_j \rangle$ 的最早开始时间 — $e(k)$;

$a_k = \langle v_i, v_j \rangle$ 的最晚开始时间 — $l(k)$;

2. 性质: ~~性质~~ $ee(v_0) = 0$;

$ee(v_j) = \max\{ee(v_i) + \text{weight}\langle v_i, v_j \rangle\}$ ($\langle v_i, v_j \rangle$ 是所有入边中的一个);

$le(v_{n-1}) = le(v_{n-1})$;

$le(v_i) = \min\{le(v_j) - \text{weight}\langle v_i, v_j \rangle\}$;

$e(k) = ee(v_i)$; ($a_k = \langle v_i, v_j \rangle$);

$l(k) = le(v_j) - \text{weight}\langle v_i, v_j \rangle$;

2° 实际的AOE网无回路, 且只有一个入度0的开始顶点, 和一个出度为0的完成顶点。

3. 算法: 各活动的最早开始时间和最晚开始时间相等, 则它属于关键路径。

内核: for ($i=0$; $i < \text{paoe} \rightarrow n$; $i++$) {

$p = \text{paoe} \rightarrow \text{vexs}[i].\text{edgelist}$;

while ($p \neq \text{NULL}$) {

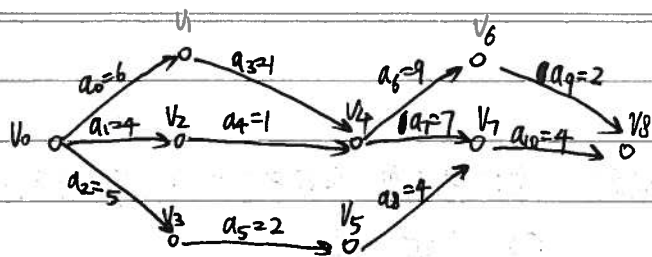
$j = p \rightarrow \text{endvex}$;

if ($e[k] == l[k]$) printf (" $\langle v_i, v_j \rangle$ ", i, j);

$p = p \rightarrow \text{nextedge}$;

}

}



i	0	1	2	3	4	5	6	7	8
$ee[i]$	0	6	4	5	7	7	16	14	18
$le[i]$	0	6	6	8	7	10	16	14	18

关键事件

关键活动可以用 $ee[k] == lc[k]$ 判定, 这里不再导出

时 $O(n^2)$