《计算机算法设计与分析》

第五章 动态规划

马丙鹏

2020年10月15日

中国科学院大学
University of Chinese Academy of Sciences

第五章 动态规划

- 5.1 一般方法
- 5.2 多段图问题
- 5.3 每对结点之间的最短路径
- 5.4 最优二分检索树
- 5.5 0/1背包问题
- 5.6 可靠性设计
- 5.7 货郎担问题
- 5.8 流水线调度问题

5.2 多段图问题

- 1. 问题的描述
  - 在多段图中求从s到t的一条最小成本的路径，可以看作是在k-2个段作出某种决策的结果。
  - 第i次决策决定V_{i+1}中的哪个结点在这条路径上，这里$1 \le i \le k-2$；
  - 最优性原理对多段图问题成立。

5.2 多段图问题

■ 2. 向前处理策略求解

□ 设P(i, j)是一条从V_i中的结点j到汇点t的最小成本路径，COST(i, j)是这条路径的成本。

V_i中的结点j到V_{i+1}中的结点l成本

□ 向前递推式

V_i中的结点j到汇点t的最小成本

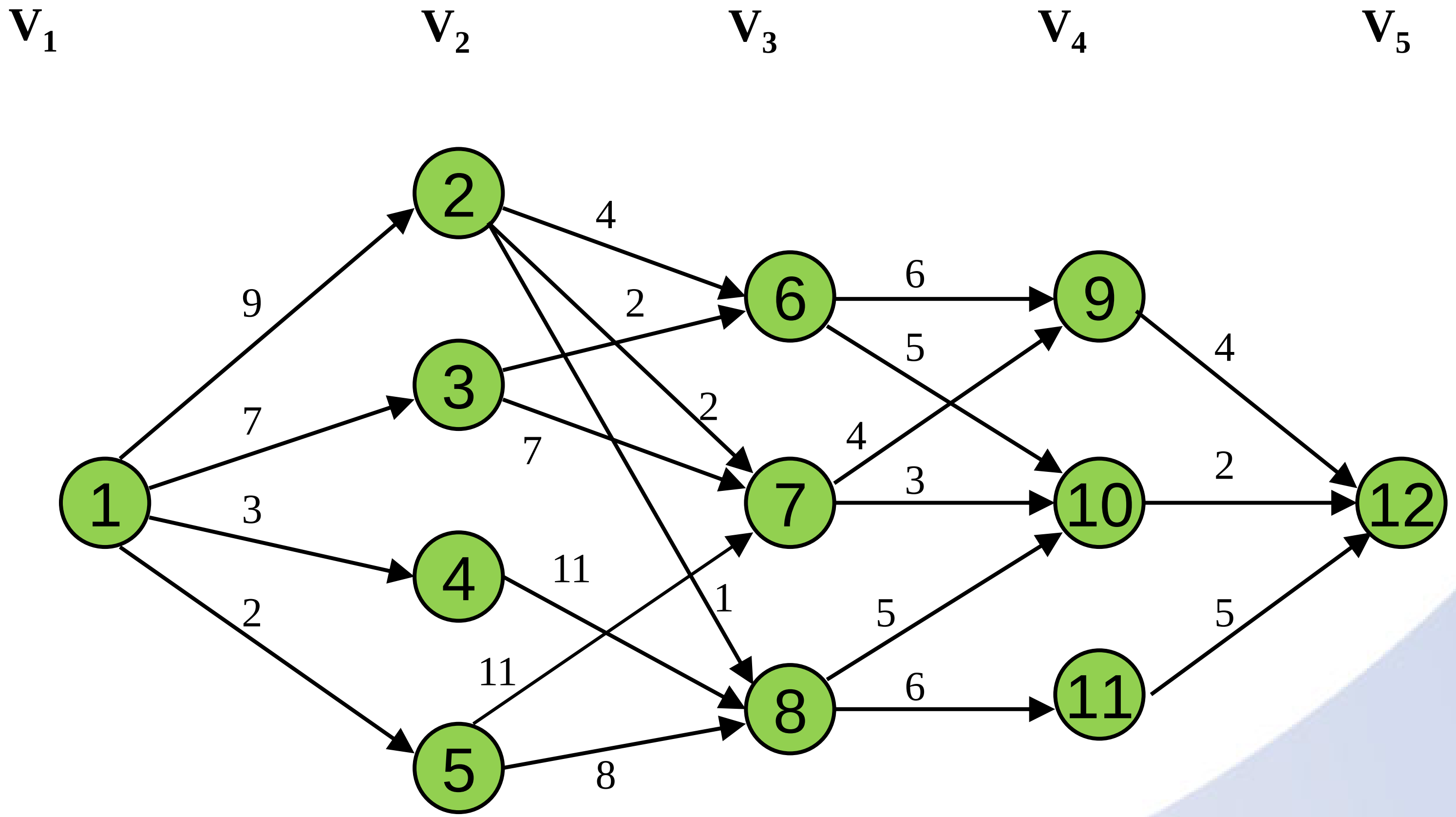
$$COST(i,j)=\min_{\substack{l\in v_{i+1}\\(j,l)\in E}}\{c(j,l)+COST(i+1,l)\}$$

V_{i+1}中的结点l到汇点t的最小成本

□ 递推过程

第k-1段

$$COST(k-1, j)=\begin{cases} c(j, t) & <j, t>\in E \\ \infty & \end{cases}$$



5段图

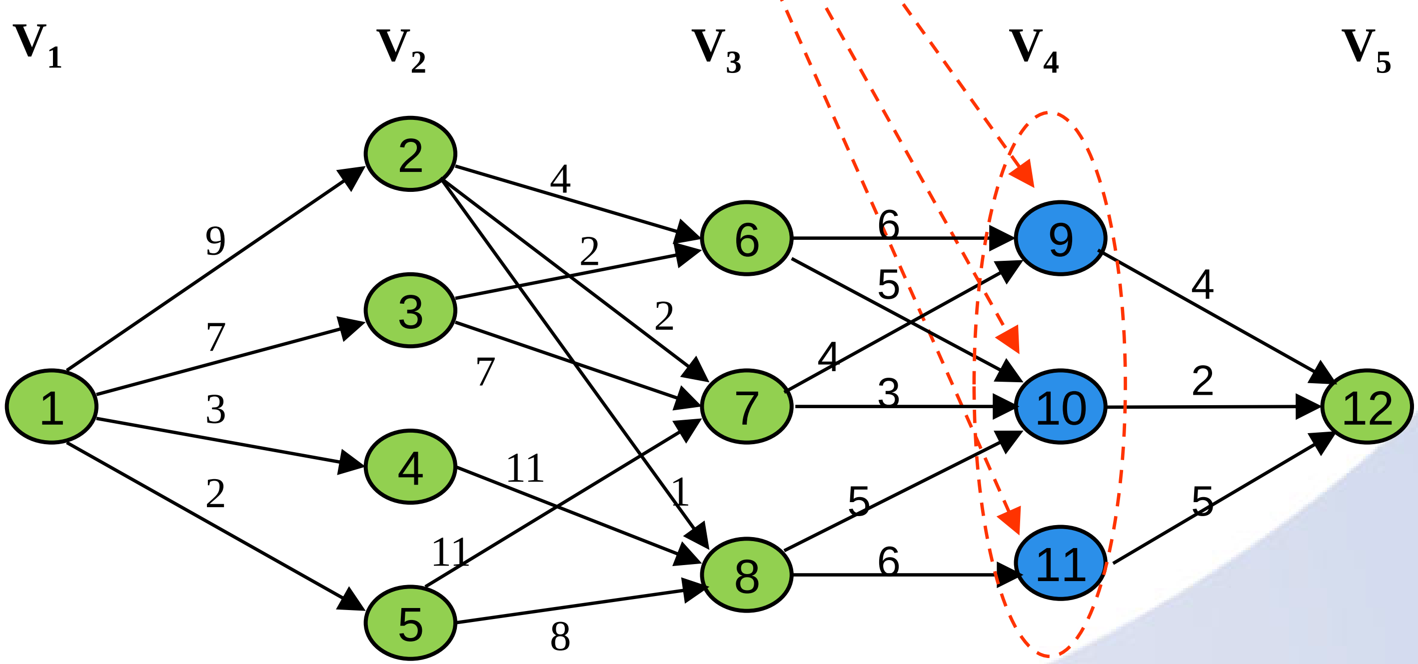
$$COST(i, j) = \min_{\substack{l \in v_{i+1} \\ <j,l> \in E}} \{c(j,l) + COST(i+1,l)\}$$

★各递推结果

第4段　COST(4, 9) = c(9, 12) = 4

COST(4, 10) = c(10, 12) = 2

COST(4, 11) = c(11, 12) = 5



5段图

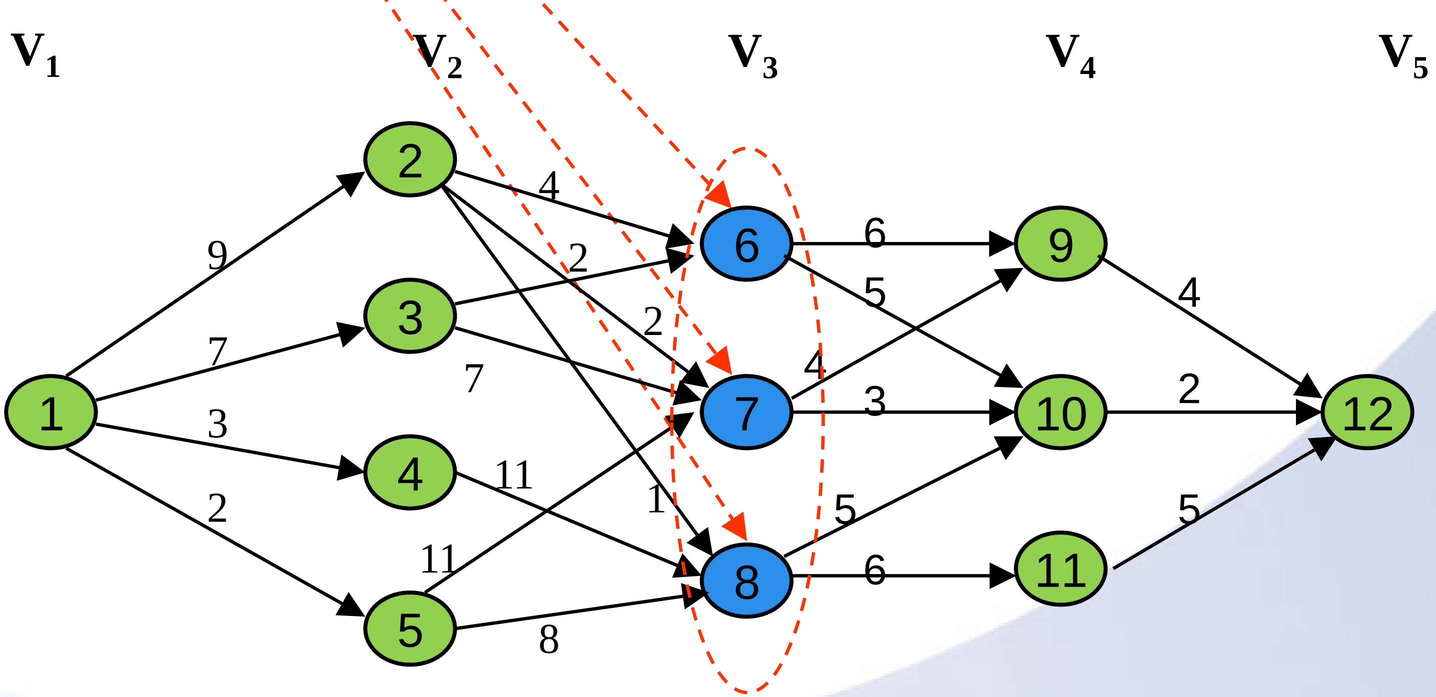
$$COST(i, j) = \min_{\substack{l \in v_{i+1} \\ <j,l> \in E}} \{c(j,l) + COST(i+1,l)\}$$

★各递推结果

第3段　COST(3, 6) = min{6+COST(4, 9), 5+COST(4, 10)} = 7

COST(3, 7) = min{4+COST(4, 9), 3+COST(4, 10)} = 5

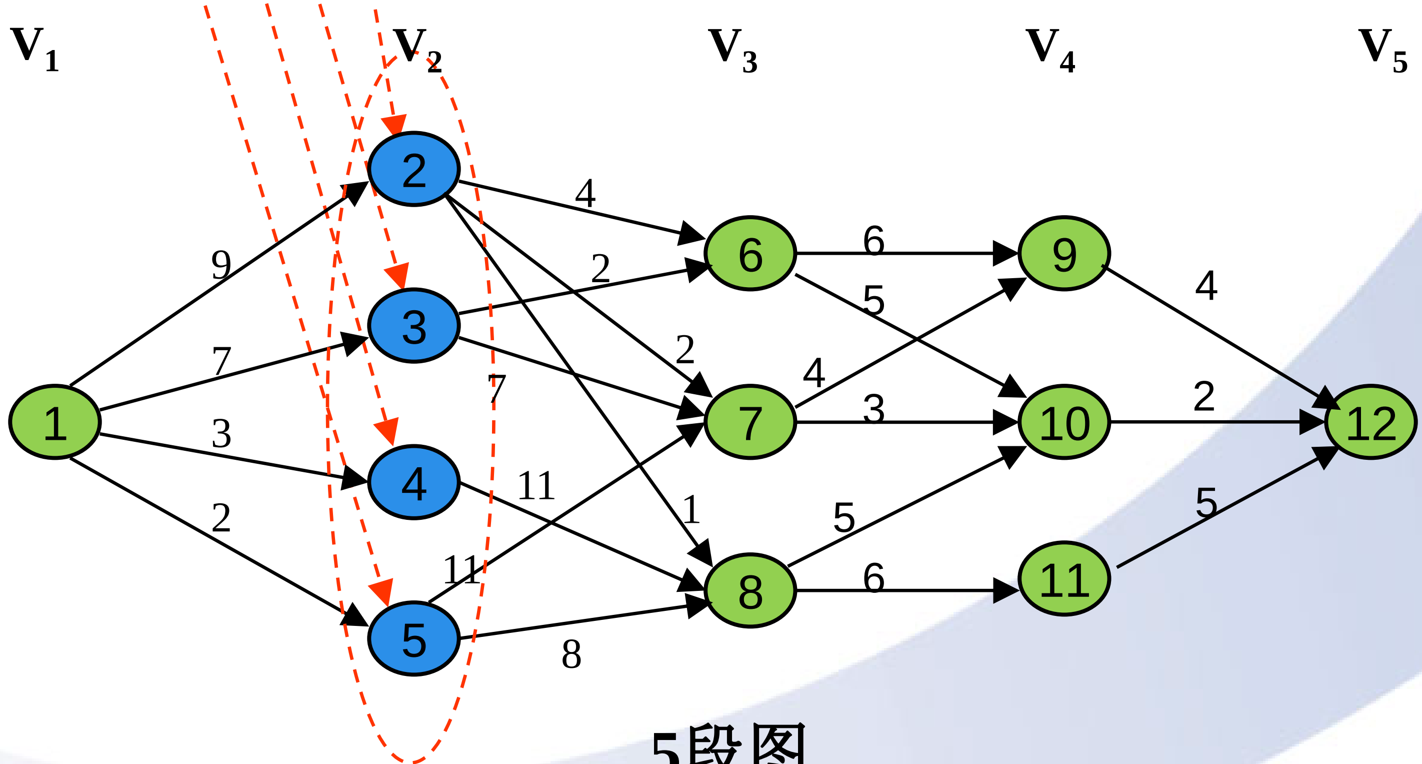
COST(3, 8) = min{5+COST(4, 10), 6+COST(4, 11)} = 7



5段图

$$COST(i, j) = \min_{\substack{l \in v_{i+1} \\ <j,l> \in E}} \{c(j,l) + COST(i+1, l)\}$$

第2段　COST(2, 2) = min{4+COST(3, 6) , 2+COST(3, 7), 1+COST(3, 8)} = 7
COST(2, 3) = 9
COST(2, 4) = 18
COST(2, 5) = 15

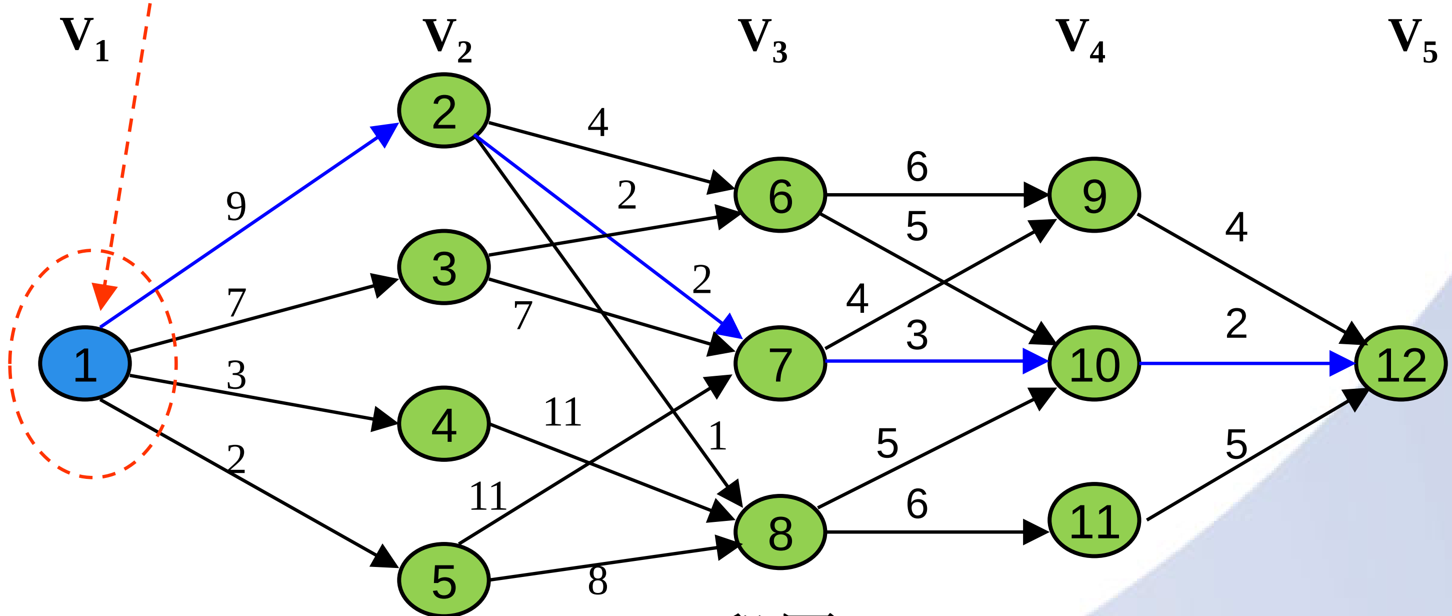


5段图

$$COST(i,j)=\min_{\substack{l\in v_{i+1}\\ <j,l>\in E}}\{c(j,l)+COST(i+1,l)\}$$

第1段 **COST(1, 1) = min{9+COST(2, 2), 7+COST(2, 3), 3+COST(2, 4), 2+COST(2, 5)}**

= 16



5段图

s到t的最小成本路径的成本 = 16

★各递推结果

第4段 COST(4, 9) = c(9, 12) = 4

COST(4, 10) = c(10, 12) = 2

COST(4, 11) = c(11, 12) = 5

第3段 COST(3, 6) = min{6+COST(4, 9), 5+COST(4, 10)} = 7

COST(3, 7) = min{4+COST(4, 9), 3+COST(4, 10)} = 5

COST(3, 8) = min{5+COST(4, 10), 6+COST(4, 11)} = 7

第2段 COST(2, 2) = min{4+COST(3, 6), 2+COST(3, 7), 1+COST(3, 8)} = 7

COST(2, 3) = 9

COST(2, 4) = 18

COST(2, 5) = 15

第1段 COST(1, 1) = min{9+COST(2, 2), 7+COST(2, 3), 3+COST(2, 4), 2+COST(2, 5)} = 16

s到t的最小成本路径的成本 = 16

★ 最小路径的求取

记 D(i, j) = 每一COST(i, j)的决策

即，使c(j, l)+COST(i+1, l)取得最小值的l值。

例：D(3, 6) = 10, D(3, 7) = 10, D(3, 8) = 10

D(2, 2) = 7, D(2, 3) = 6, D(2, 4) = 8, D(2, 5) = 8

D(1, 1) = 2

根据D(1,1)的决策值向后递推求取最小成本路径：

- v_2 = D(1, 1)=2
- v_3 = D(2, D(1, 1)) = 7
- v_4 = D(3, D(2, D(1, 1))) = D(3, 7) = 10

故由s到t的最小成本路径是：1→2→7→10→12

$$COST(i,j)=\min_{\substack{l\in v_{i+1}\\ <j,l>\in E}}\{c(j,l)+COST(i+1,l)\}$$

第一段

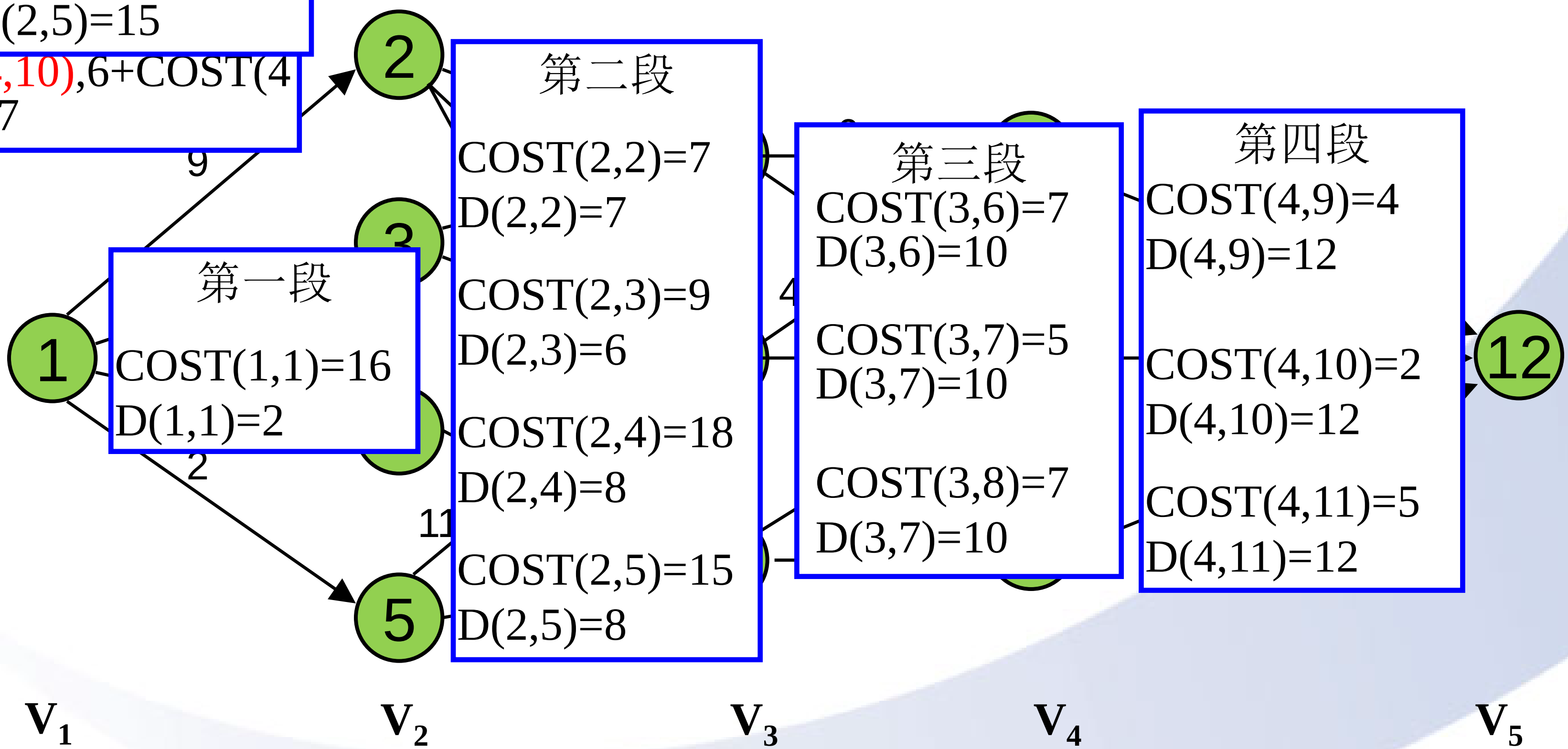
COST(1,1)=min{9+COST(2,2),7+COST(2,3),3+COST(2,4),2+COST(2,5)}=16

COST(2,4)=18

COST(2,5)=15

OST(4,10),6+COST(4,11)}=7

s到t的最小成本路径的成本： 16



第一段

COST(1,1)=16
D(1,1)=2

第二段

COST(2,2)=7
D(2,2)=7

COST(2,3)=9
D(2,3)=6

COST(2,4)=18
D(2,4)=8

COST(2,5)=15
D(2,5)=8

第三段

COST(3,6)=7
D(3,6)=10

COST(3,7)=5
D(3,7)=10

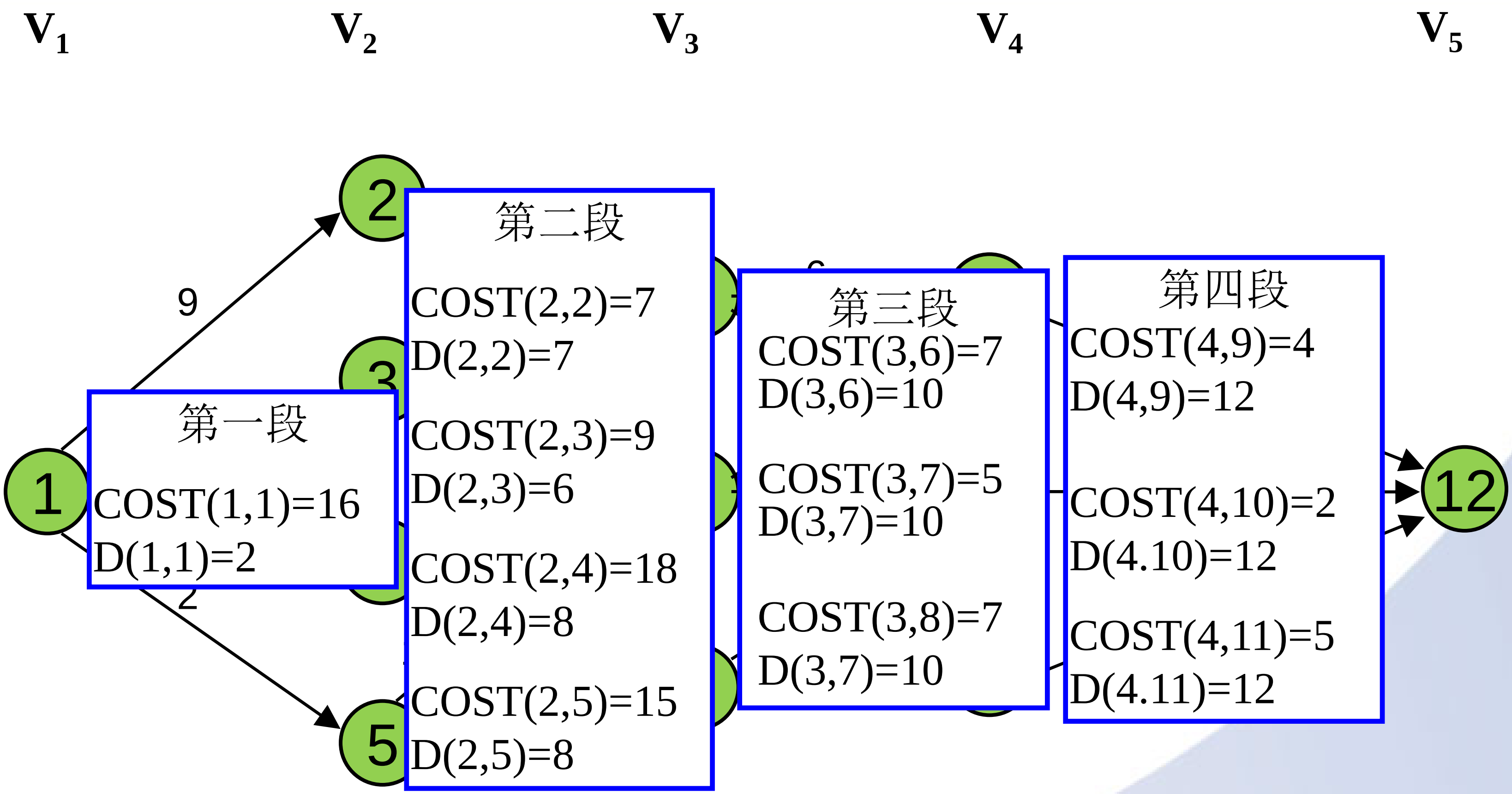
COST(3,8)=7
D(3,7)=10

第四段

COST(4,9)=4
D(4,9)=12

COST(4,10)=2
D(4,10)=12

COST(4,11)=5
D(4,11)=12


V1
V2
V3
V4
V5
1
2
5
12
9
2
第一段
COST(1,1)=16
D(1,1)=2
第二段
COST(2,2)=7
D(2,2)=7
COST(2,3)=9
D(2,3)=6
COST(2,4)=18
D(2,4)=8
COST(2,5)=15
D(2,5)=8
第三段
COST(3,6)=7
D(3,6)=10
COST(3,7)=5
D(3,7)=10
COST(3,8)=7
D(3,7)=10
第四段
COST(4,9)=4
D(4,9)=12
COST(4,10)=2
D(4.10)=12
COST(4,11)=5
D(4.11)=12

5.2 多段图问题

- 2. 向前处理策略求解
  - 算法描述
    - 结点的编号规则
      源点s编号为1，然后依次对$V_2, V_3, \ldots, V_{k-1}$中的结点编号，汇点t编号为n。
    - 目的
      使对COST和D的计算仅按n-1, n-2, …, 1的次序计算即可，
      无需考虑标示结点所在段的第一个下标。

算法5.1 多段图的向前处理算法

```
procedure FGRAPH(E, k, n, P)
  //输入是按段的顺序给结点编号的，有n个结点的k段图。E是边
   集，c(i, j)是边<i, j>的成本。P(1:k)带出最小成本路径//
  real COST(n); integer D(n-1), P(k), r, j, k, n
  COST(n)←0
  for j←n-1 to 1 by -1 do //计算COST(j)//
     设r是具有性质：<j, r>∈E且使c(j, r)+COST(r)取最小值的结点
     COST(j)←c(j, r) + COST(r)
     D(j) ←r   //记录决策值//
  repeat
  P(1)←1; P(k)←n
  for j←2 to k-1 do    //找路径上的第j个结点//
     P(j) ←D(P(j-1))   //回溯求出该路径//
  repeat
end FGRAPH
```

寻找第j个结点到终点的最短路径

$\Theta(n+e)$

第j个结点到汇点的最短路径上的下一个节点

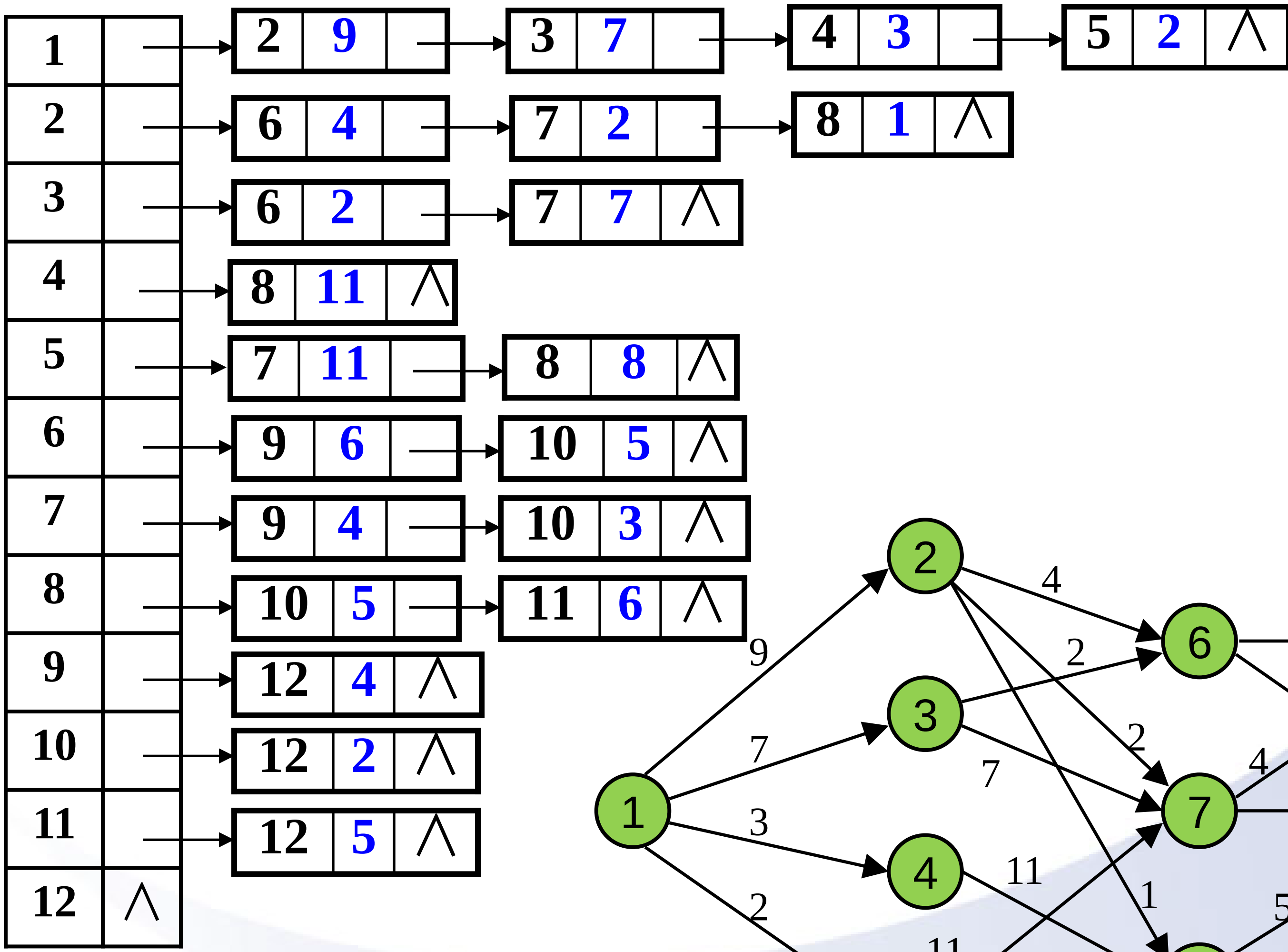
5.2 多段图问题

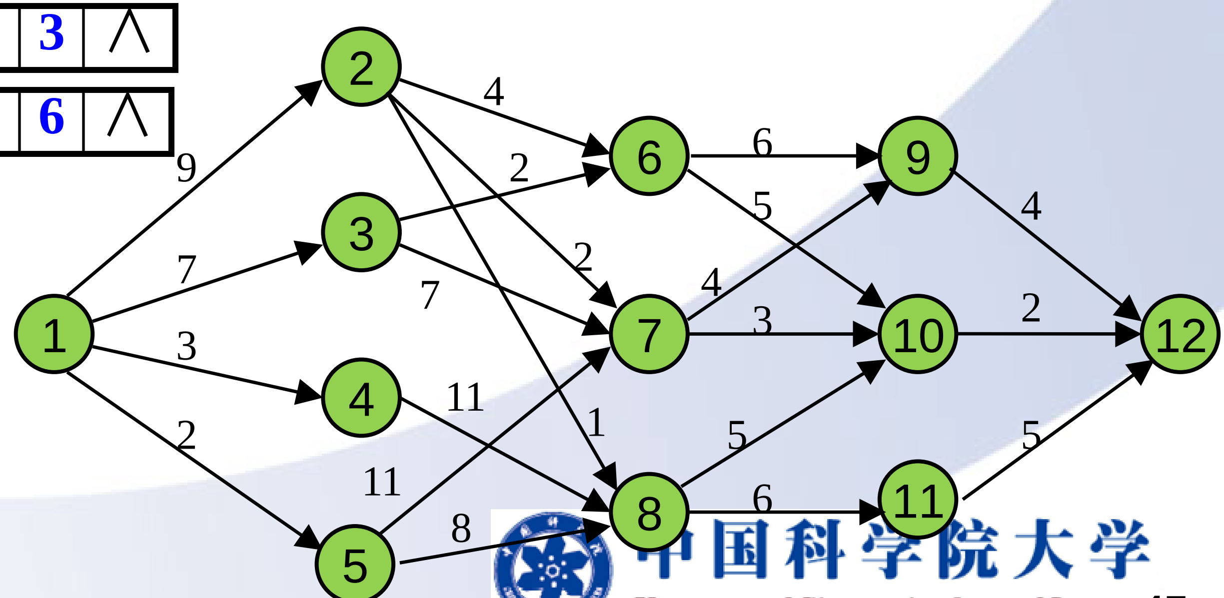
- 2. 向前处理策略求解
  - 算法的时间复杂度
    - 若G采用邻接表表示，总计算时间为：

$$\Theta(n+e)$$

    - 邻接表: 邻接表是图的一种链式存储结构, 对图中的每个顶点建立一个单链表, 链表中的结点有3个域, 分别存储顶点, 边的成本和下一个结点的指针.

5.2 多段图问题


1 → 2 9 → 3 7 → 4 3 → 5 2 ∧
2 → 6 4 → 7 2 → 8 1 ∧
3 → 6 2 → 7 7 ∧
4 → 8 11 ∧
5 → 7 11 → 8 8 ∧
6 → 9 6 → 10 5 ∧
7 → 9 4 → 10 3 ∧
8 → 10 5 → 11 6 ∧
9 → 12 4 ∧
10 → 12 2 ∧
11 → 12 5 ∧
12 ∧


1
2
3
4
5
6
7
8
9
10
11
12
9
7
3
2
4
2
1
2
7
11
11
8
6
5
4
3
5
6
4
2
5

算法的执行过程

k=5; n=12;

COST(12)=0;

for j=11 downto 1 do

{ COST(j)=min{c(j, r)+COST(r)};

D(j)=r; }

P(1)=1; P(k)=12;

for j=2 to 4 do P(j)=D(P(j−1)) ;

COST(11)=5 D(11)=12
COST(10)=2 D(10)=12
COST(9)=4 D(9)=12

COST(8)=7 D(8)=10
COST(7)=5 D(7)=10
COST(6)=7 D(6)=10

COST(5)=15 D(5)=8
COST(4)=18 D(4)=8
COST(3)=9 D(3)=6
COST(2)=7 D(2)=7

COST

1	2	3	4	5	6	7	8	9	10	11	12
16	7	9	18	15	7	5	7	4	2	5	0

COST(1)=16

D(1)=2

D

1	2	3	4	5	6	7	8	9	10	11
2	7	6	8	8	10	10	10	12	12	12

P

1	2	3	4	5
1	2	7	10	12

中国科学院大学
University of Chinese Academy of Sciences

5.2 多段图问题

■ 3. 向后处理策略求解

- 设BP(i, j)是一条从源点s到V_i中的结点j的最小成本路径，BCOST(i, j)是这条路径的成本。
- 向后递推式

源点s到V_i中的结点j的最小成本

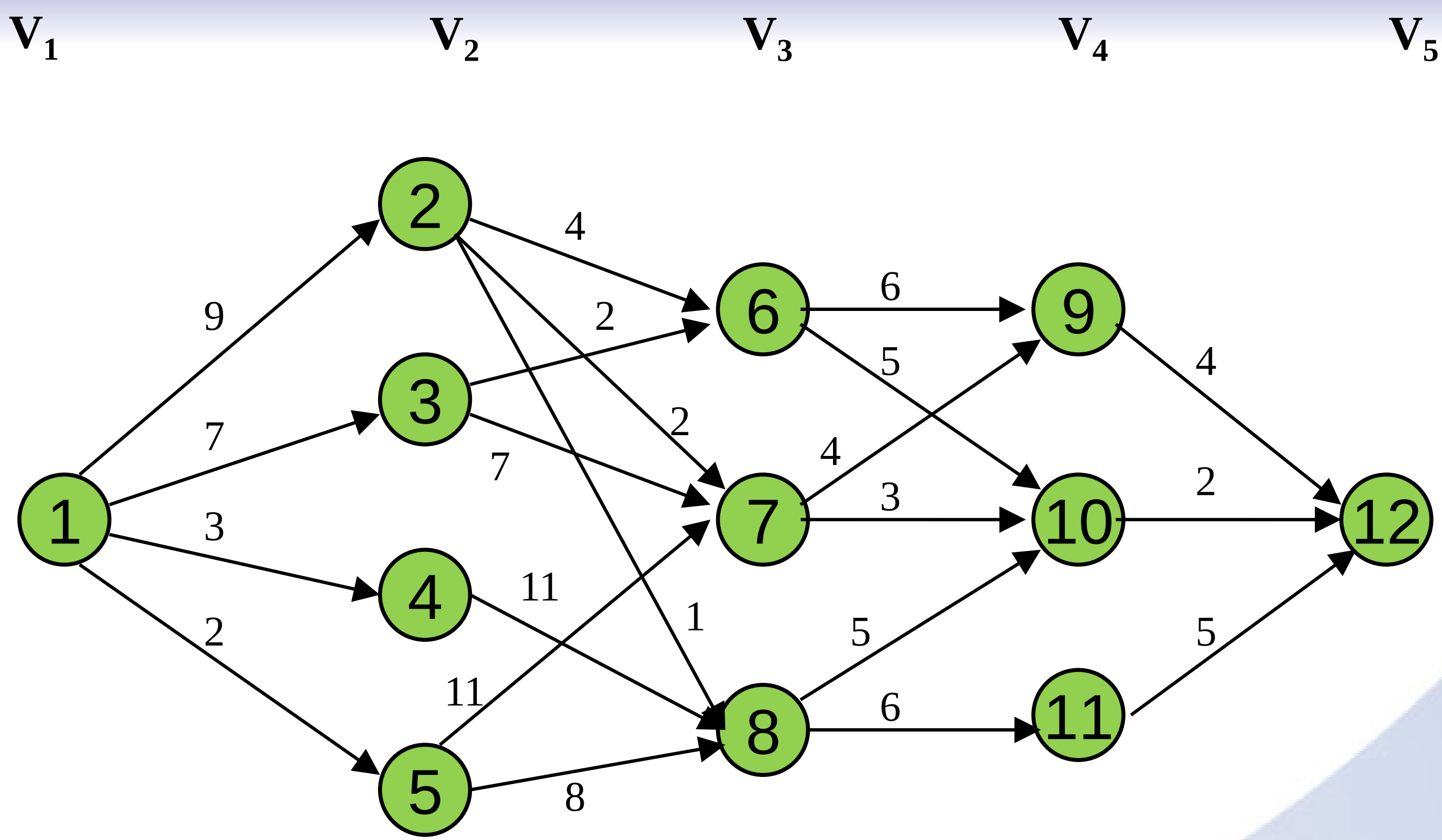
源点s到V_{i-1}中的结点l的最小成本

V_{i-1}中的结点l 到V_{i+1}中的结点j的成本

$$BCOST(i, j) = \min_{\substack{l \in v_{i-1} \\ (l, j) \in E}} \{BCOST(i-1, l) + c(l, j)\}$$

- 递推过程
  - 第2段

$$\mathrm{BCOST}(2, j) = \begin{cases} c(1, j) & <1, j> \in E \\ \infty & \end{cases}$$



5段图

$$BCOST(i,j)=\min_{\substack{l\in v_{i-1}\\(l,j)\in E}}\{BCOST(i-1,l)+c(l,j)\}$$

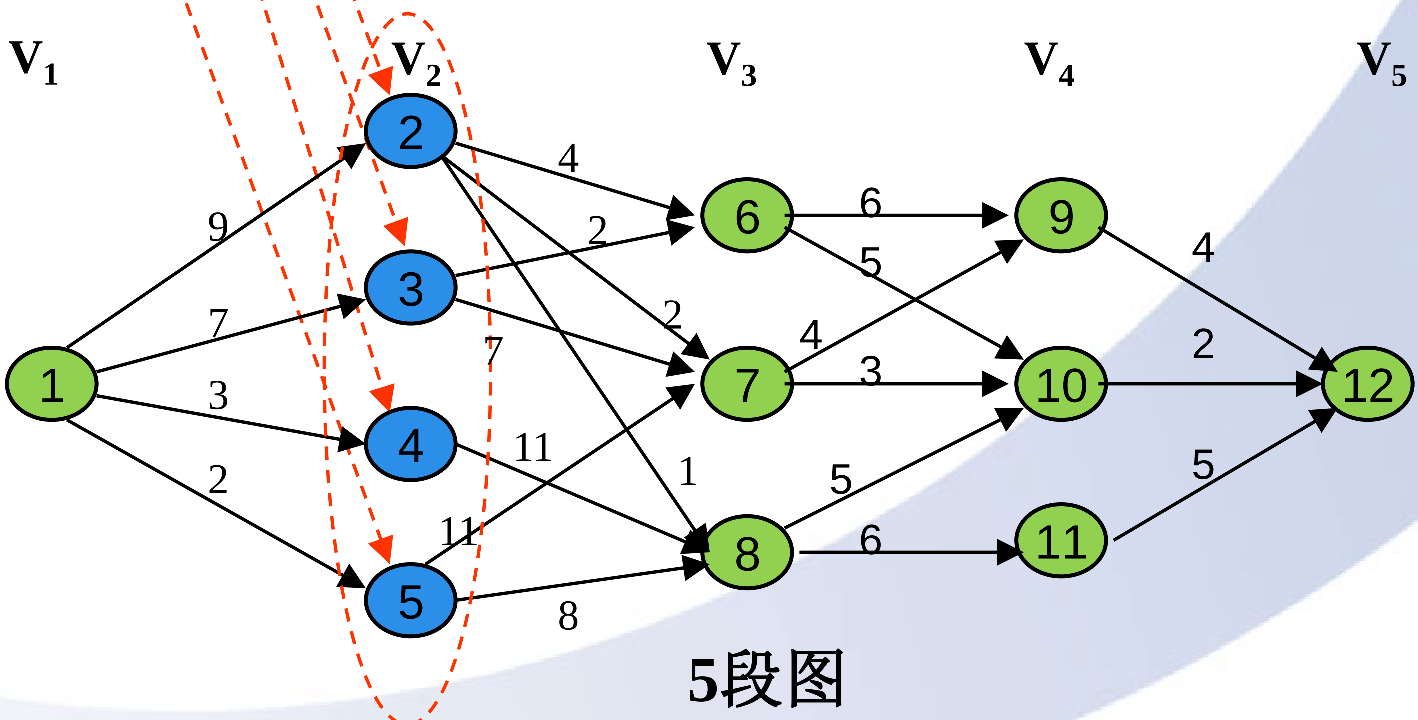
★各递推结果

第2段　**BCOST(2, 2) = 9**

BCOST(2, 3) = 7

BCOST(2, 4) = 3

BCOST(2, 5) = 2



5段图

中国科学院大学
University of Chinese Academy of Sciences

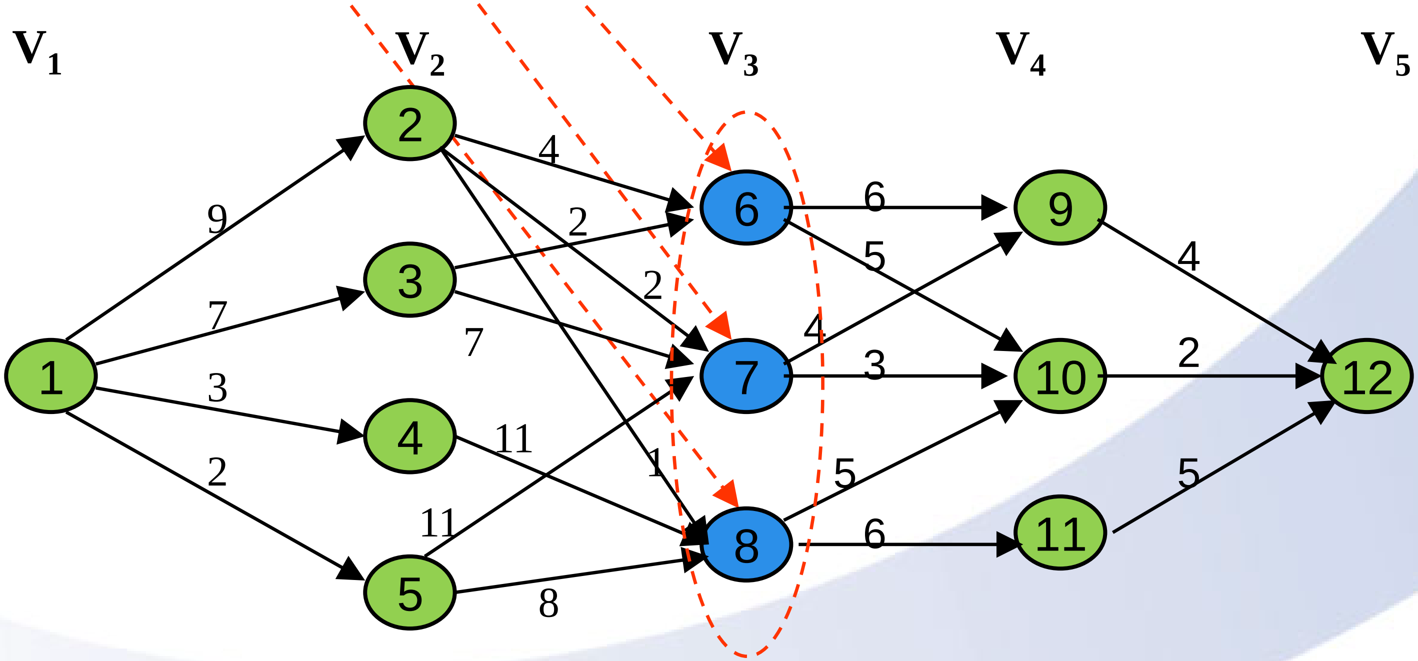
$$BCOST(i,j)=\min_{\substack{l\in v_{i-1}\\(l,j)\in E}}\{BCOST(i-1,l)+c(l,j)\}$$

★各递推结果

第3段 BCOST(3, 6) = min{BCOST(2, 2)+4, BCOST(2, 3)+2} = 9

BCOST(3, 7) = min{BCOST(2, 2)+2, BCOST(2, 3)+7, BCOST(2, 5)+11} = 11

BCOST(3, 8) = min{BCOST(2, 2)+1, BCOST(2, 4)+11, BCOST(2, 5)+8} = 10



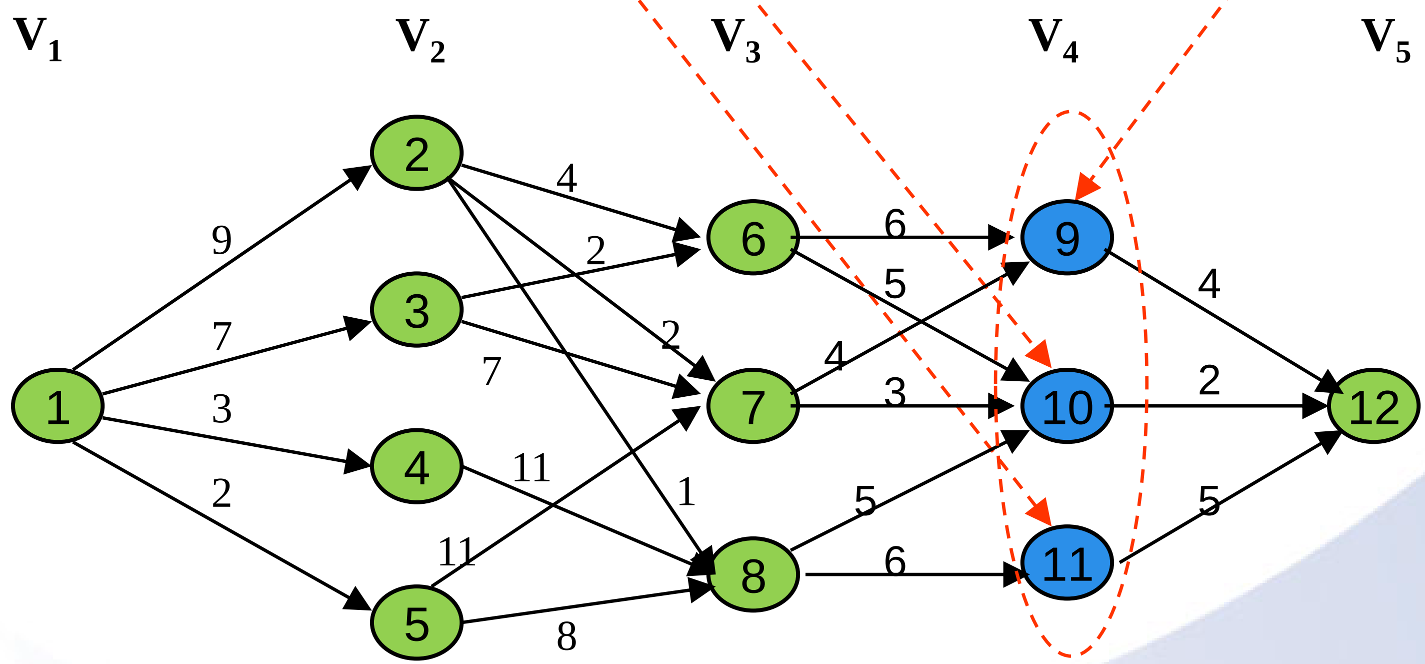
5段图

$$BCOST(i, j) = \min_{\substack{l \in v_{i-1} \\ (l, j) \in E}} \{BCOST(i-1, l) + c(l, j)\}$$

第4段 BCOST(4, 9) = min{BCOST(3, 6)+6, BCOST(3, 7)+4} = 15

BCOST(4, 10) = min{BCOST(3, 6)+5, BCOST(3, 7)+3, BCOST(3, 8)+5} = 14

BCOST(4, 11) = min{BCOST(3, 8)+6} = 16

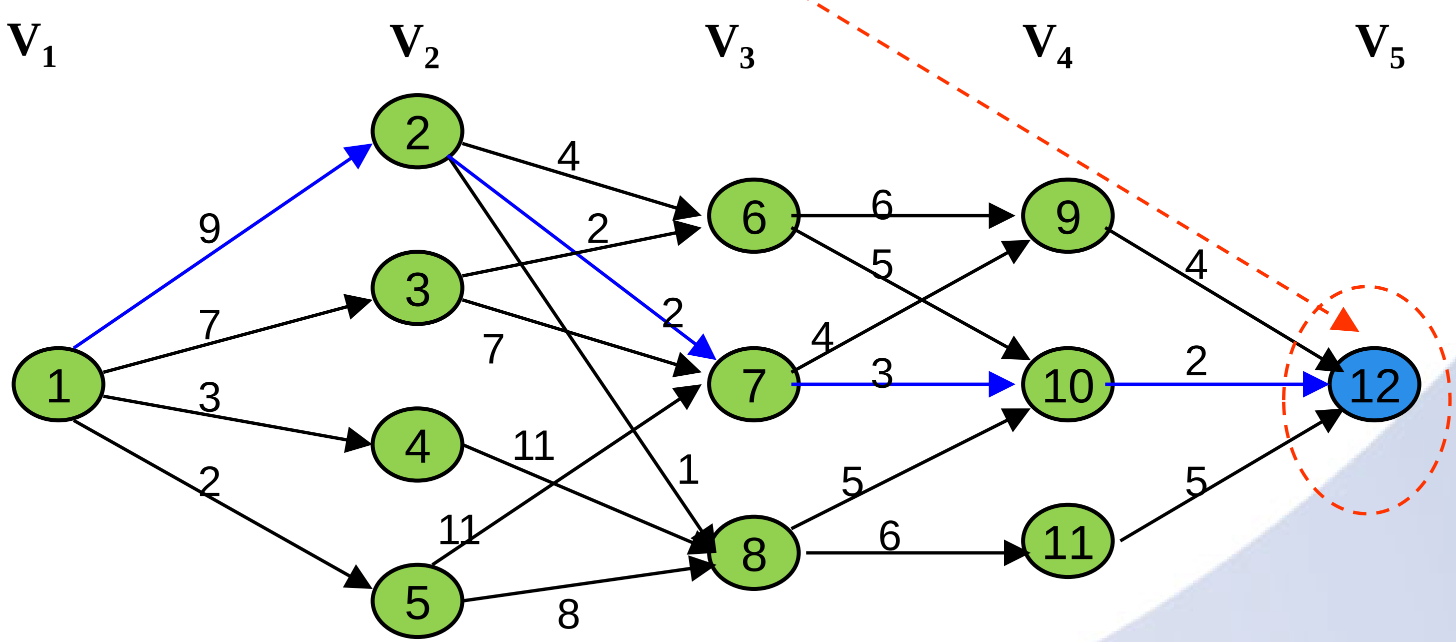


5段图

$$BCOST(i,j)=\min_{\substack{l\in v_{i-1}\\ (l,j)\in E}}\{BCOST(i-1,l)+c(l,j)\}$$

第5段　　BCOST(5, 12) = min{BCOST(4, 9)+4, BCOST(4, 10)+2, BCOST(4, 11)+5}

= 16



s到t的最小成本路径的成本 = 16

★各递推结果

第2段 BCOST(2, 2) = 9 BCOST(2, 3) = 7

BCOST(2, 4) = 3 BCOST(2, 5) = 2

第3段 BCOST(3, 6) = min{BCOST(2, 2)+4, BCOST(2, 3)+2} = 9

BCOST(3, 7) = min{BCOST(2, 2)+2, BCOST(2, 3)+7, BCOST(2, 5)+11} = 11

BCOST(3, 8) = min{BCOST(2, 4)+11, BCOST(2, 5)+8} = 10

第4段 BCOST(4, 9) = min{BCOST(3, 6)+6, BCOST(3, 7)+4} = 15

BCOST(4, 10) = min{BCOST(3, 6)+5, BCOST(3, 7)+3, BCOST(3, 8)+5} = 14

BCOST(4, 11) = min{BCOST(3, 8)+6} = 16

第5段 BCOST(5, 12) = min{BCOST(4, 9)+4, BCOST(4, 10)+2, BCOST(4, 11)+5}

= 16

s到t的最小成本路径的成本 = 16

★ 最小路径的求取

记 BD(i, j) = 每一COST(i, j)的决策

即，使COST(i-1, l)+c(l, j)取得最小值的l值。

例：BD(3, 6) = 3, BD(3, 7) = 2, BD(3, 8) = 5

BD(4, 9) = 6, BD(4, 10) = 7, BD(4, 11) = 8

BD(5, 12) = 10

根据D(5, 12)的决策值向前递推求取最小成本路径：

- v_4 = BD(5, 12) =10
- v_3 = BD(4, BD(5, 12)) = 7
- v_2 = BD(3, BD(4, BD(5, 12))) = BD(3, 7) = 2

故由s到t的最小成本路径是：1→2→7→10→12

$$BCOST(i,j)=\min_{\substack{l\in v_{i-1}\\(l,j)\in E}}\{COST(i-1,l)+c(l,$$

s到t的最小成本路径的成本： 16

第四段

BCOST(5,12)=min{BCOST(4,9)+4,BCOST(4,10)+2,BCOST(4,11)+5}=16

第三段

BCOST(4,9)=min{BCOST(... BCOS... }=15

BCOST(4,10)=14

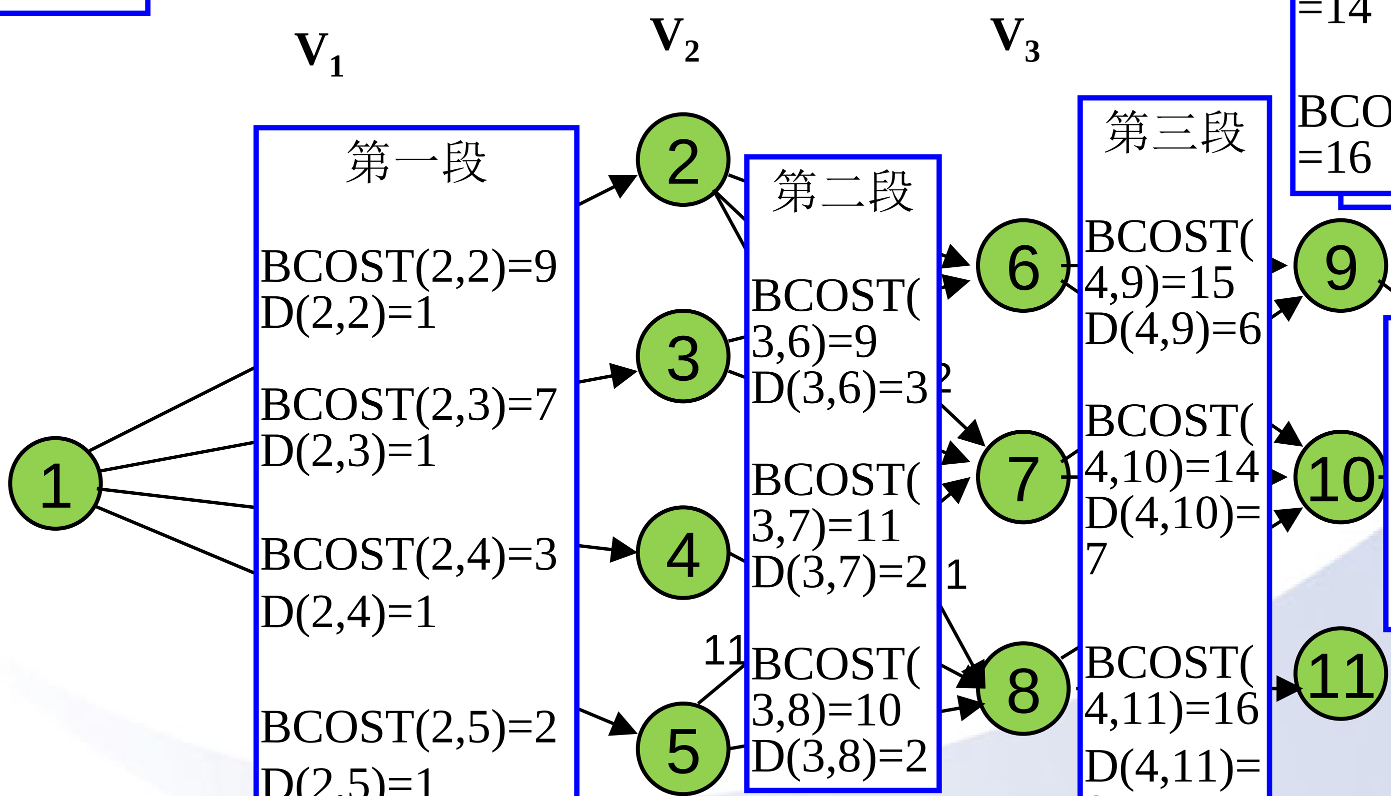
BCOST(4,11)=16

第二段

...T(3,6) BCOS... +4,BC... ,3)+2}

...T(3,7) BCOS... +2,BC... ,3)+7, ...T(2,5) 11

BCOST(3,8)=min{BCOS... +1,BC... ,4)+11, ...T(2,5) 0



第一段

BCOST(2,2)=9
D(2,2)=1

BCOST(2,3)=7
D(2,3)=1

BCOST(2,4)=3
D(2,4)=1

BCOST(2,5)=2
D(2,5)=1

第二段

BCOST(3,6)=9
D(3,6)=3

BCOST(3,7)=11
D(3,7)=2

BCOST(3,8)=10
D(3,8)=2

第三段

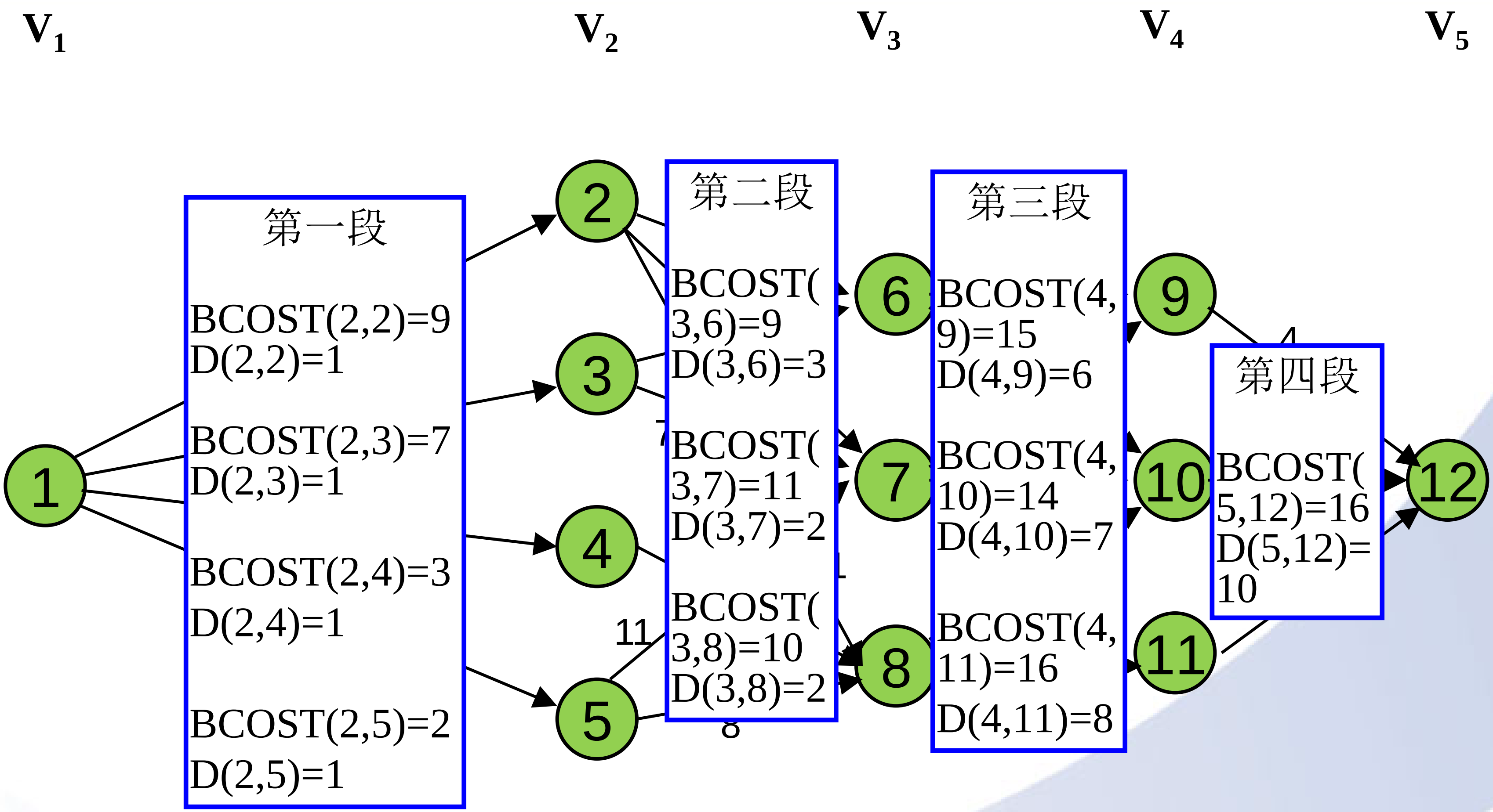
BCOST(4,9)=15
D(4,9)=6

BCOST(4,10)=14
D(4,10)=7

BCOST(4,11)=16
D(4,11)=8

第四段

BCOST(5,12)=16
D(5,12)=10

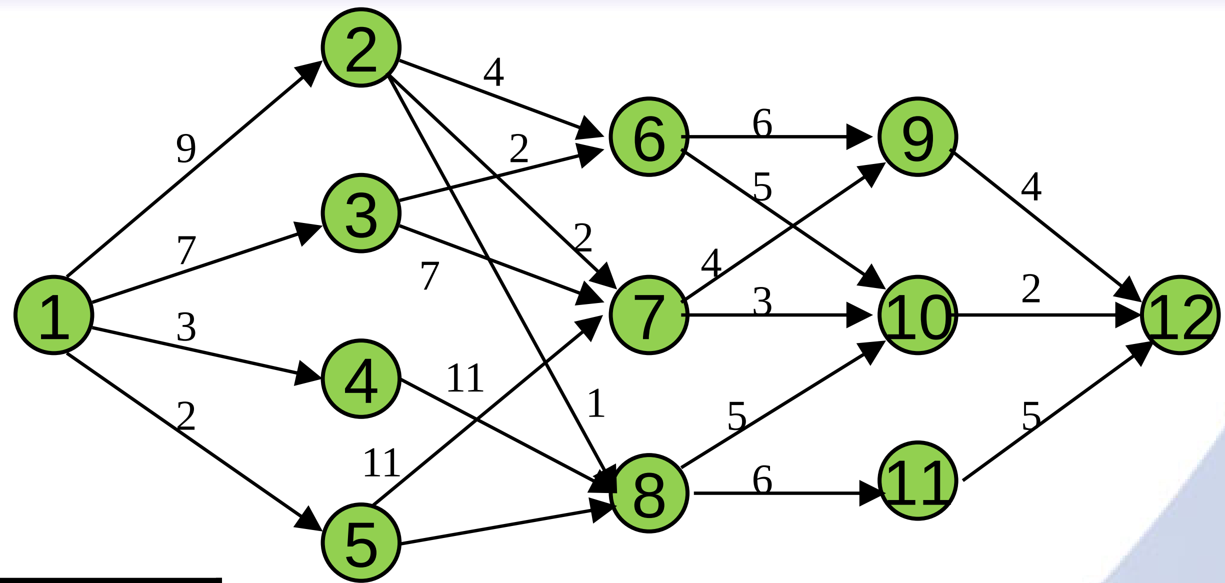


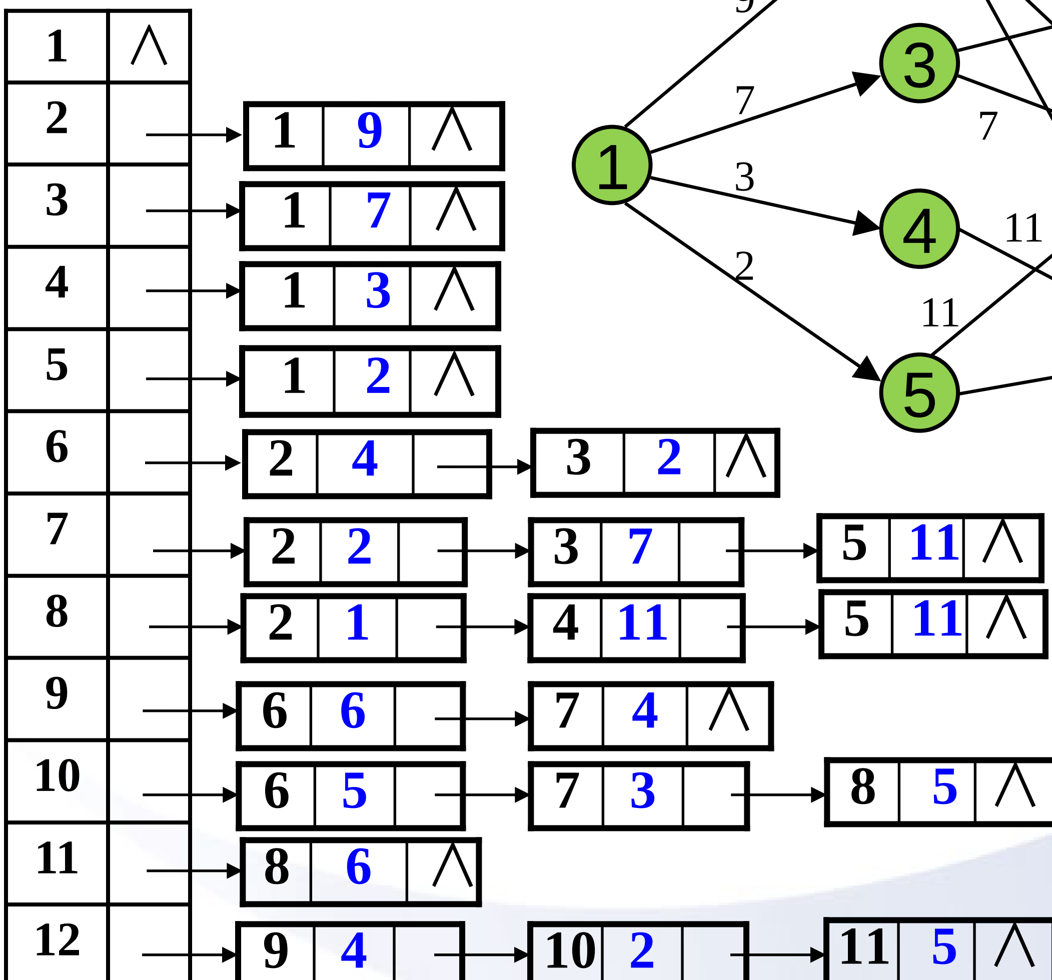
V1
V2
V3
V4
V5
第一段
BCOST(2,2)=9
D(2,2)=1
BCOST(2,3)=7
D(2,3)=1
BCOST(2,4)=3
D(2,4)=1
BCOST(2,5)=2
D(2,5)=1
第二段
BCOST(3,6)=9
D(3,6)=3
BCOST(3,7)=11
D(3,7)=2
BCOST(3,8)=10
D(3,8)=2
第三段
BCOST(4,9)=15
D(4,9)=6
BCOST(4,10)=14
D(4,10)=7
BCOST(4,11)=16
D(4,11)=8
第四段
BCOST(5,12)=16
D(5,12)=10
1
2
3
4
5
6
7
8
9
10
11
12

算法5.2 多段图的向后处理算法

```
procedure BGRAPH(E, k, n, P)
  //输入是按段的顺序给结点编号的，有n个结点的k段图。E是边
   集，c(i, j)是边<i, j>的成本。P(1:k)带出最小成本路径//
  real BCOST(n); integer BD(n-1), P(k), r, j, k, n
  BCOST(1)←0
  for j←2  to n  do //计算BCOST(j)//
     设r是具有<r, j>∈E且使BCOST(r)+ c(r, j)取最小值性质的结点
     BCOST(j)← BCOST(r)+ c(r, j)
     BD(j) ←r    //记录决策值//
  repeat
  P(1)←1; P(k)←n
  for j←k-1 to 2 by -1 do     //找路径上的第j个结点//
      P(j) ←D(P(j+1))    //回溯求出该路径//
  repeat
end BGRAPH
```

5.2 多段图问题





对于向后处理算法多段图用逆邻接表来存储

第五章 动态规划

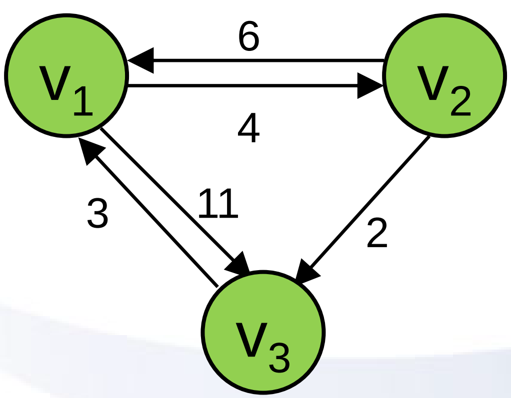
- 5.1 一般方法
- 5.2 多段图问题
- 5.3 每对结点之间的最短路径
- 5.4 最优二分检索树
- 5.5 0/1背包问题
- 5.6 可靠性设计
- 5.7 货郎担问题
- 5.8 流水线调度问题

5.3 每对结点之间的最短路径

■ 1. 问题描述

- 设G=(V, E)是一个有n个结点的有向图，C是G的成本邻接矩阵，C中元素有：

$$c(i, j)=\begin{cases} 0, & i = j \\ \text{边}\langle i, j\rangle\text{的成本}, & i\neq j\text{且}\langle i, j\rangle\in E(G) \\ \infty, & i\neq j\text{且其中}1\leq i, j\leq n \end{cases}$$

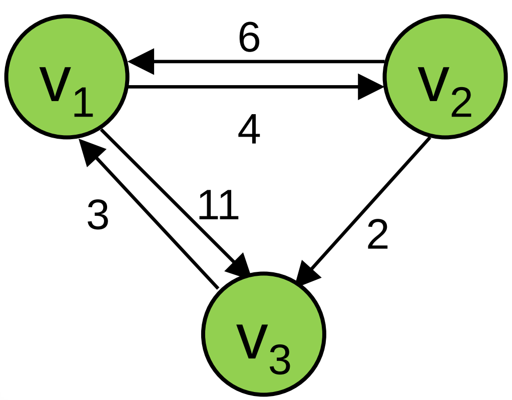


	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

5.3 每对结点之间的最短路径

■ 1. 问题描述

- □每对结点之间的最短路径问题：
  - ➢求满足下述条件的矩阵A，A中的任一元素A(i, j)代表结点i到结点j的最短路径长度。



	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

C：成本邻接矩阵

	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

A：每对结点间最短路径矩阵

5.3 每对结点之间的最短路径

- **1. 问题描述**
  - 利用单源最短路径算法求解
    - 计算n个结点的单源最短路径。
    - 时间复杂度：$O(n^3)$。
  - 利用动态规划策略求解
    - 将求解G中每对结点之间的最短路径问题转化成一个多阶段决策过程。
    - 最优性原理对于该问题是否成立？
    - 决策什么？

5.3 每对结点之间的最短路径

即证明最短路径具有最优性原理刻画的性质

■ 2. 动态规划求解策略

□最优性原理对于每对结点之间的最短路径问题成立

对G的一条由i到j的最短路径(假设该路径中不包含环)，设k是该路径上的一个中间结点：

i, ..., k, …, j

由i到k的最短路径　k是中间结点　由k到j的最短路径

则，由i到k和k到j的两条子路径将分别是由i到k和由k到j的最短路径。否则i, ..., k, …, j也将不是由i到j的最短路径。

故，最优性原理对于该问题成立。

中国科学院大学
University of Chinese Academy of Sciences

5.3 每对结点之间的最短路径

■ 2. 动态规划求解策略

□ 多阶段决策过程

假设所有n个结点依次有从1到n的编号。

设k是由i到j的最短路径上编号最高的中间结点：

i, ..., k, …, j

k是编号最高的中间结点

则由i到k的子路径上将不会有比编号k-1更大的结点；同理，由k到j的子路径上也将不会有比编号k-1更大的结点。

构造多阶段决策过程：对由i到j的最短路径，首先决策哪一个结点是该路径上具有最大编号的中间结点k，然后再去求取由i到k和由k到j的最短路径——其中应不包含比k-1还大的中间结点。

5.3 每对结点之间的最短路径

- **2. 动态规划求解策略**
  - 递推关系式
    - 记$A^k(i, j)$表示从i到j并且不经过比k还大的结点的最短路径长度。
    - $A^0(i, j)$：i至j的路径上中间不包含任何中间结点
    - $A^1(i, j)$：i至j的路径上可以包含中间结点，但仅能是结点1
    - $A^2(i, j)$：i至j的路径上中间结点可以是结点1, 2
    - $A^3(i, j)$：i至j的路径上中间结点可以是结点1, 2, 3
    - …

5.3 每对结点之间的最短路径

- **2. 动态规划求解策略**
  - 递推关系式
    - $A^n(i, j)$：i至j的路径上中间可以包含结点1, 2, …, n(i, j除外)
    - 因为所有结点的编号不会大于n，所以$A(i, j) = A^n(i, j)$，即由i到j的最短路径不通过编号比n还大的结点。
    - 所有的A(i, j) 构成结果矩阵A。

中国科学院大学
University of Chinese Academy of Sciences

5.3 每对结点之间的最短路径

■ 2. 动态规划求解策略

□递推关系式

➢注：该路径可以经过结点n，也可以不经过结点n。

✓若该路径不经过结点n，则

$A^{n}(i, j) = A^{n-1}(i, j)$

✓若该路径经过结点n，则

$A^{n}(i, j) = A^{n-1}(i, n) + A^{n-1}(n, j)$

➢故可得

$$A^{n}(i, j) = \min\{\underbrace{A^{n-1}(i, j)}_{\text{不经过n结点}}, \underbrace{A^{n-1}(i, n) + A^{n-1}(n, j)}_{\text{经过n结点}}\}$$

5.3 每对结点之间的最短路径

- 2. 动态规划求解策略
  - 递推关系式
    - 注：对任意的k，k≥1，有，

不经过k结点

经过k结点

$$A^k(i, j) = \min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^{k-1}(k, j)\}$$

- 初值：

$$A^0(i, j) = C(i, j),\ 1 \le i \le n,\ 1 \le j \le n$$

- 递推计算：

$$A^1(i, j) = \min\{A^0(i, j), A^0(i, 1) + A^0(1, j)\}$$

$$A^2(i, j) = \min\{A^1(i, j), A^1(i, 2) + A^1(2, j)\}$$

$$\ldots$$

$$A^n(i, j) = \min\{A^{n-1}(i, j), A^{n-1}(i, n) + A^{n-1}(n, j)\}$$

$$= A(i, j)$$

算法5.3 每对结点之间的最短路径长度

```
procedure ALL-PATHS(COST, A, n)
  //COST(n,n)是n结点图的成本邻接矩阵；A(i, j)是结点vi到vj的最短路
   径的成本；COST(i, i)=0, 1≤i≤n//
  integer i, j, k, n; real COST(n, n), A(n, n)
  for i←1 to n do
      for j←1 to n do
          A(i, j) ←COST(i, j)   //用COST(i, j)对A⁰赋初值//
      repeat
  repeat
  for k←1 to n do
      for i←1 to n do
          for j←1 to n do
                  A(i, j) ← min{A(i, j), A(i, k) + A(k, j)}
          repeat
      repeat
  repeat
end ALL-PATHS
```

$$A^0(i, j)$$

$$A^1(i, j), A^2(i, j), \ldots, A^n(i, j)=A(i, j)$$

$$A^k(i, j) = \min\{A^{k-1}(i, j), A^{k-1}(i, k)+A^{k-1}(k, j)\}$$

5.3 每对结点之间的最短路径

■ 3. 算法描述

□程序正确性讨论

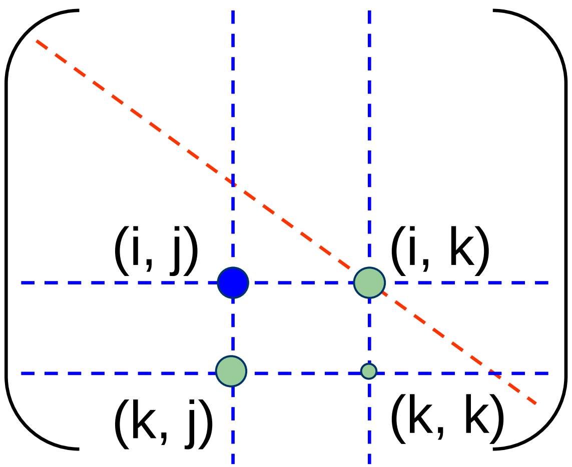
➢∵ (1)在第k-1到第k次的迭代过程中，A的第k行、第k列元素不变，即

$$A^k(i, k) = A^{k-1}(i, k)$$

$$A^k(k, j) = A^{k-1}(k, j)$$

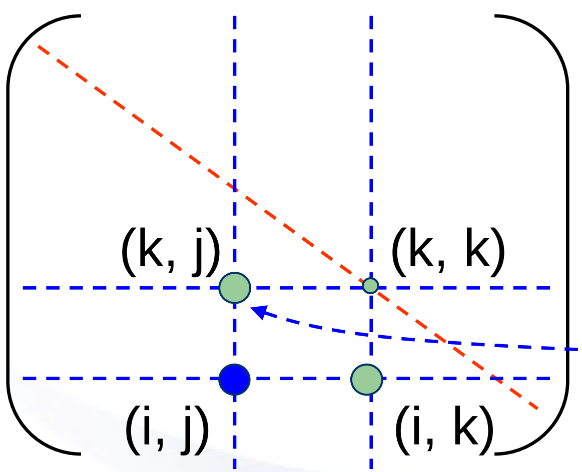
➢(2)讨论k与i, j的关系，可知A(i, j), A(i, k)和A(k, j)执行的先后次序

5.3 每对结点之间的最短路径



k>j且k>i, 则有

$A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^{k-1}(k, j)\}$

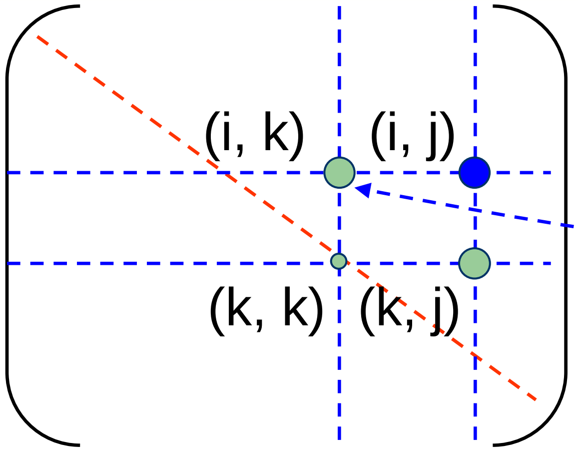


k>j且k<i, 则有

$A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^k(k, j)\}$

$A^{k-1}(k, j)$

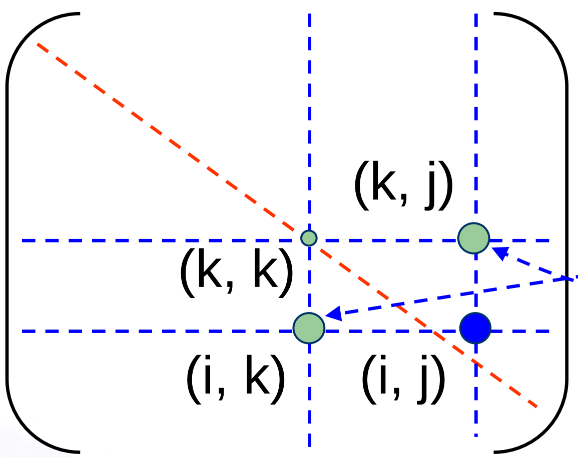
5.3 每对结点之间的最短路径



k<j且k>i , 则有

$$A^{k}(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k}(i, k) + A^{k-1}(k, j)\}$$

$$A^{k}(i, k) \Downarrow A^{k-1}(i, k)$$



k<j且k<i, 则有

$$A^{k}(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k}(i, k) + A^{k}(k, j)\}$$

$$A^{k}(i, k) \Downarrow A^{k-1}(i, k) \qquad A^{k}(k, j) \Downarrow A^{k-1}(k, j)$$

5.3 每对结点之间的最短路径

■ 3. 算法描述

□程序正确性讨论

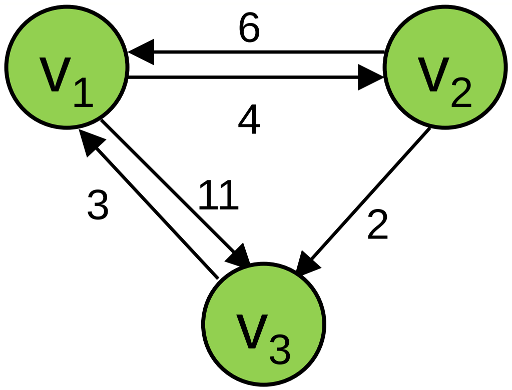
$$\left.\begin{array}{l} A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k}(i, k) + A^{k-1}(k, j)\} \\ A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^{k}(k, j)\} \\ A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k}(i, k) + A^{k}(k, j)\} \end{array}\right\}$$

$$\equiv A^k(i, j) \leftarrow \min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^{k-1}(k, j)\}$$

∴ 在算法的计算过程中取消了A的上标，并保证了每次计算的$A^k(i, j)$即为

$$\min\{A^{k-1}(i, j), A^{k-1}(i, k) + A^{k-1}(k, j)\}$$

例5.8 有向图如图所示



求图中所有结点间的最短路径矩阵A:

A^0	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A^1	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

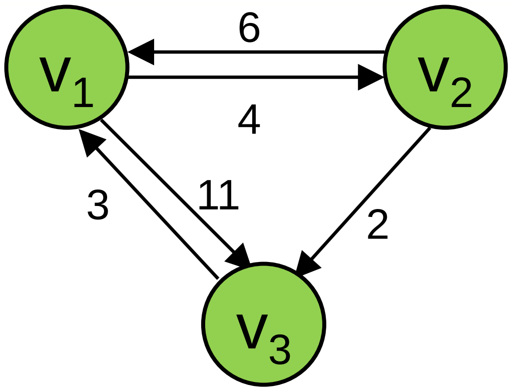
	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A^2	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

A^3	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

注： $A(i, j) = \infty$表明G中从i到j没有有向路径

例5.8 有向图如图所示



A^0	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A^1	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

算法的设计说明(1)：

$$A^k(i, k) = A^{k-1}(i, k)$$

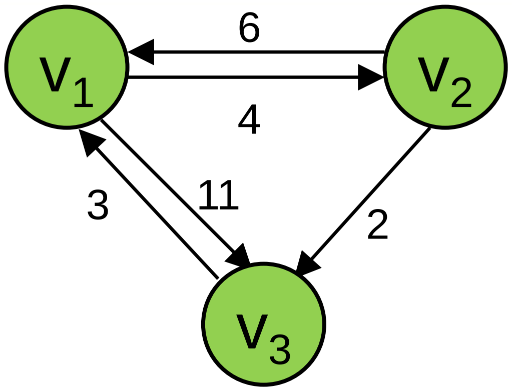
$$A^k(k, j) = A^{k-1}(k, j)$$

即：在由第k-1到第k次的迭代过程中，A的第k行、第k列元素保持不变

$$A^1(2, 3)=\min\{A^0(2, 3), A^0(2, 1)+A^0(1, 3)\}$$
$$=\min\{2, 6+11\}$$
$$=2$$

$$A^1(3, 2)=\min\{A^0(3, 2), A^0(3, 1)+A^0(1, 2)\}$$
$$=\min\{\infty, 3+4\}$$
$$=7$$

例5.8 有向图如图所示



A^0	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A^1	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

$A^2(1, 3)=\min\{A^1(1, 3), A^1(1, 2)+A^1(2, 3)\}$

$=\min\{11, 4+2\}$

$=6$

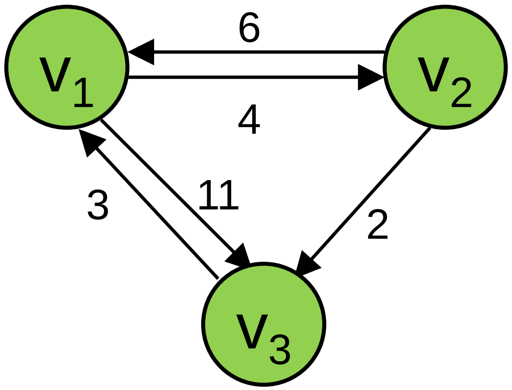
A^2	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

$A^2(3, 1)=\min\{A^1(3, 1), A^1(3, 2)+A^1(2, 1)\}$

$=\min\{3, 7+6\}$

$=3$

例5.8 有向图如图所示



A^0	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A^1	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

A^2	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

A^3	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

$$A^3(1, 2)=\min\{A^2(1, 2), A^2(1, 3)+A^2(3, 2)\}$$
$$=\min\{4, 6+7\}$$
$$=4$$

$$A^3(2, 1)=\min\{A^2(2, 1), A^2(2, 3)+A^2(3, 1)\}$$
$$=\min\{6, 2+3\}$$
$$=5$$

5.3 每对结点之间的最短路径

■ 3. 算法描述

□性能分析

➢计算时间 $=\Theta(n^3)$

➢ 注：该时间与A的值无关：

```
for k←1 to n do                         迭代n次
    for i←1 to n do                     迭代n次
        for j←1 to n do                 迭代n次
            A(i, j) ← min{A(i, j), A(i, k) + A(k, j)}
        repeat
    repeat
repeat
```

中国科学院大学
University of Chinese Academy of Sciences

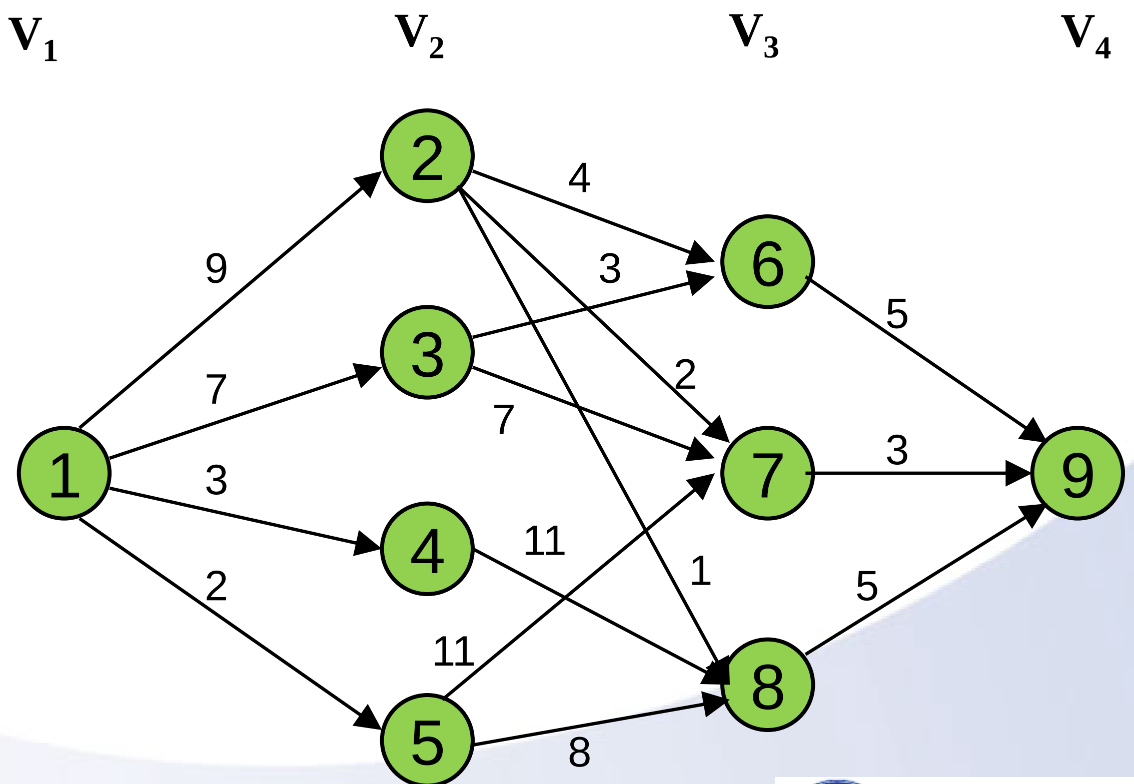
5.3 每对结点之间的最短路径

- 3. 算法描述
  - ∞的处理
    - 设M是E中最大成本的一条边的成本，则 $A^n(i, j) \le (n-1)*M$。
    - 因此，对于成本邻接矩阵中的∞用一个大于(n-1)*M的值代替。
    - 如果在算法结束时，$A(i, j) > (n-1)*M$，则表明G中没有由i到j的有向路径。

作业-课后练习13

■ 问题描述

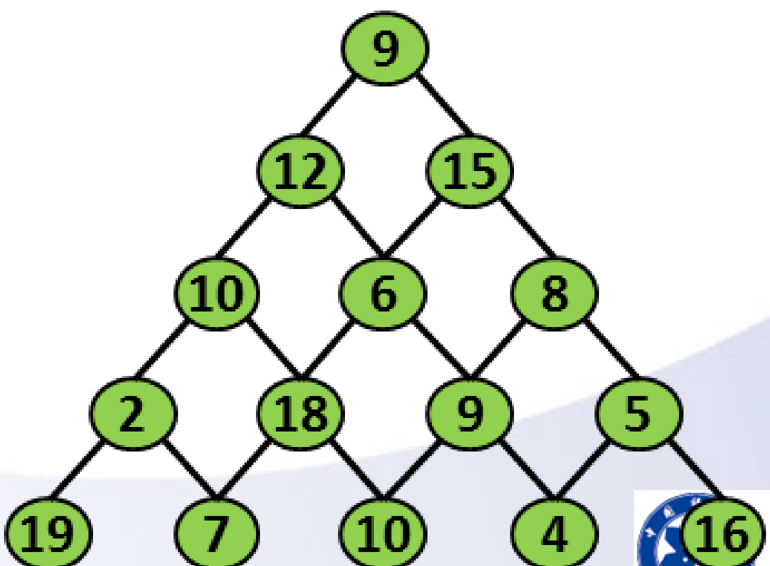
□用向前递推方法求解下面的四段图问题。



4段图

作业-课后练习14

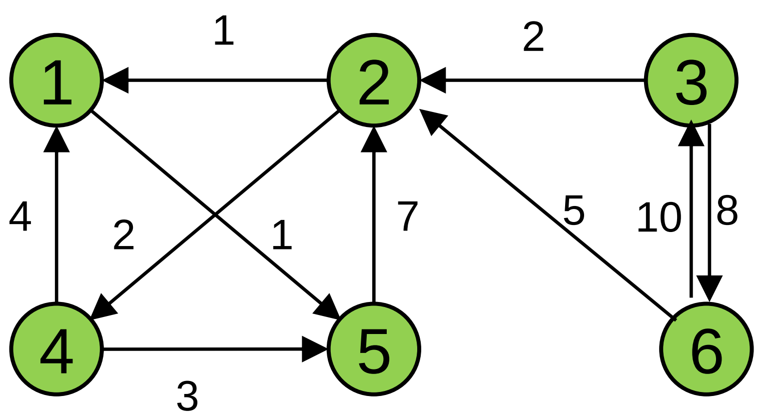
- 问题描述
  - 给定一个树塔，如下图所示。在此树塔中，从顶部出发，可以选择向左走还是向右走，一直走到最低层。请找出一条路径,使得路径上的数值和最大，并给出该数值和。(15分)
  - 2014年本课程的考试试题



作业-课后练习15

■ 问题描述

- □求下面两个图里面每对结点之间的最短路径



End