

## 《数据结构》题解规范

说明：本题解规范提供给同学们参考，希望同学们做题规范。由于时间仓促，若规范中存在不妥之处，请同学们告诉我们。

——石家庄铁道学院《数据结构》课程组

### 第 3 章 栈和队列

一、现有一中缀表达式  $(13+5) \times 3 - 65 / (5+2 \times 4) + 7$ ，请采用栈结构将其转换为后缀表达式并计算结果，要求画出栈的变化情况。

解答过程：中缀表达式转换为后缀表达式：

步骤	中缀表达式	STACK	输出
1	$(13+5) \times 3 - 65 / (5+2 \times 4) + 7 \#$	#	
2	$13+5) \times 3 - 65 / (5+2 \times 4) + 7 \#$	# (	
3	$+5) \times 3 - 65 / (5+2 \times 4) + 7 \#$	# (	13
4	$5) \times 3 - 65 / (5+2 \times 4) + 7 \#$	# ( +	13
5	$) \times 3 - 65 / (5+2 \times 4) + 7 \#$	# ( +	13 5
6	$\times 3 - 65 / (5+2 \times 4) + 7 \#$	#	13 5 +
7	$3 - 65 / (5+2 \times 4) + 7 \#$	# ×	13 5 +
8	$- 65 / (5+2 \times 4) + 7 \#$	# ×	13 5 + 3
9	$- 65 / (5+2 \times 4) + 7 \#$	#	13 5 + 3 ×
10	$65 / (5+2 \times 4) + 7 \#$	# -	13 5 + 3 ×
11	$/ (5+2 \times 4) + 7 \#$	# -	13 5 + 3 × 65
12	$(5+2 \times 4) + 7 \#$	# - /	13 5 + 3 × 65
13	$5+2 \times 4) + 7 \#$	# - / (	13 5 + 3 × 65
14	$+2 \times 4) + 7 \#$	# - / (	13 5 + 3 × 65 5
15	$2 \times 4) + 7 \#$	# - / ( +	13 5 + 3 × 65 5
16	$\times 4) + 7 \#$	# - / ( +	13 5 + 3 × 65 5 2
17	$4) + 7 \#$	# - / ( + ×	13 5 + 3 × 65 5 2

18	) + 7 #	# - / ( + ×	13 5 + 3 × 65 5 2 4
19	+ 7 #	# - /	13 5 + 3 × 65 5 2 4 × +
20	+ 7 #	# -	13 5 + 3 × 65 5 2 4 × + /
21	+ 7 #	#	13 5 + 3 × 65 5 2 4 × + / -
22	7 #	# +	13 5 + 3 × 65 5 2 4 × + / -
23	#	# +	13 5 + 3 × 65 5 2 4 × + / - 7
24	#	#	13 5 + 3 × 65 5 2 4 × + / - 7 +

后缀表达式的计算：

步骤	后缀表达式	STACK
1	13 5 + 3 × 65 5 2 4 × + / - 7 + #	#
2	5 + 3 × 65 5 2 4 × + / - 7 + #	#13
3	+ 3 × 65 5 2 4 × + / - 7 + #	#13 5
4	3 × 65 5 2 4 × + / - 7 + #	#18
5	× 65 5 2 4 × + / - 7 + #	#18 3
6	65 5 2 4 × + / - 7 + #	#54
7	5 2 4 × + / - 7 + #	#54 65
8	2 4 × + / - 7 + #	#54 65 5
9	4 × + / - 7 + #	#54 65 5 2
10	× + / - 7 + #	#54 65 5 2 4
11	+ / - 7 + #	#54 65 5 8
12	/ - 7 + #	#54 65 13
13	- 7 + #	#54 5
14	7 + #	#49
15	+ #	#49 7
16	#	#56

最后的计算结果为 56。

## 第4章 串

二、已知模式串“aaaab”，请按照 KMP 算法给出该模式串各字符的 next 值和 next 修正值。

解答过程：

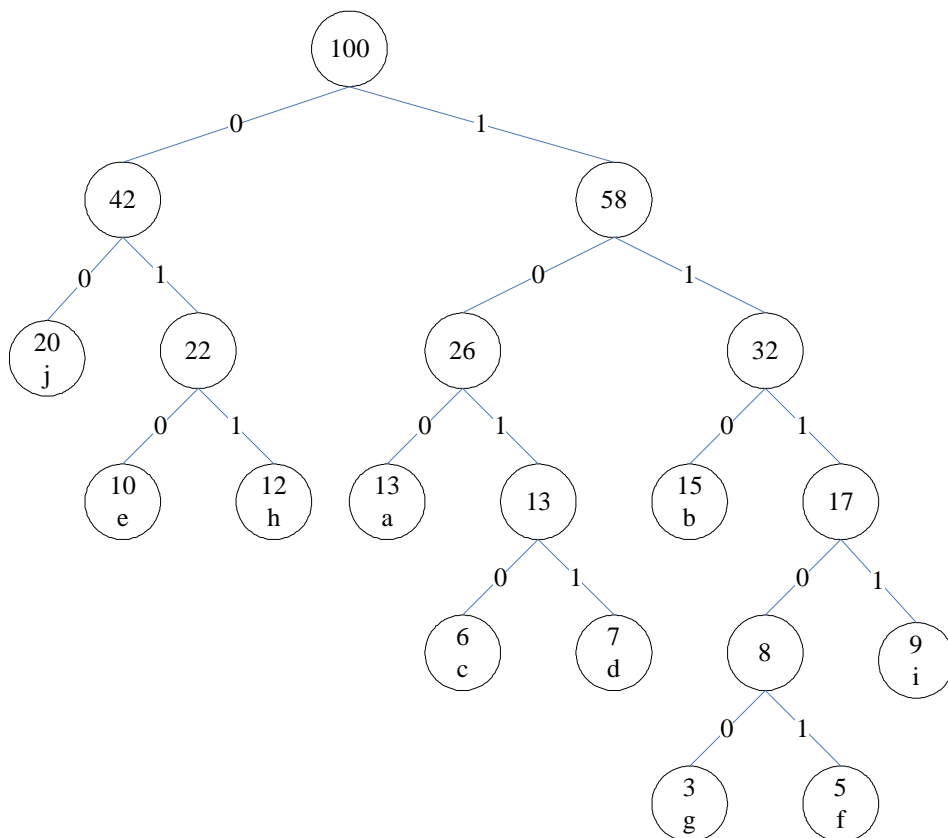
j	1	2	3	4	5
模式	a	a	a	a	b
next[j]	0	1	2	3	4
修正 next[j]	0	0	0	0	4

## 第6章 树和二叉树

三、已知通讯中传输的字符 a, b, c, d, e, f, g, h, i, j 的传输频率分别为 13, 15, 6, 7, 10, 5, 3, 12, 9, 20。构造对应的哈夫曼树，并给出各字符相对应的哈夫曼编码。

解答过程：

构造的哈夫曼树如下：



若采用的编码方案为左支为 0，右支为 1，则各字符的哈夫曼编码如下：

<b>a 100</b>	<b>d 1011</b>	<b>g 11100</b>	<b>j 00</b>
<b>b 110</b>	<b>e 010</b>	<b>h 011</b>	
<b>c 1010</b>	<b>f 11101</b>	<b>i 1111</b>	

四、已知森林的先序遍历序列和中序遍历序列，请画出该森林，要求给出求解过程。

森林的先序遍历序列：ABECDFGHIJ

森林的中序遍历序列：EBCDAFHIGJ

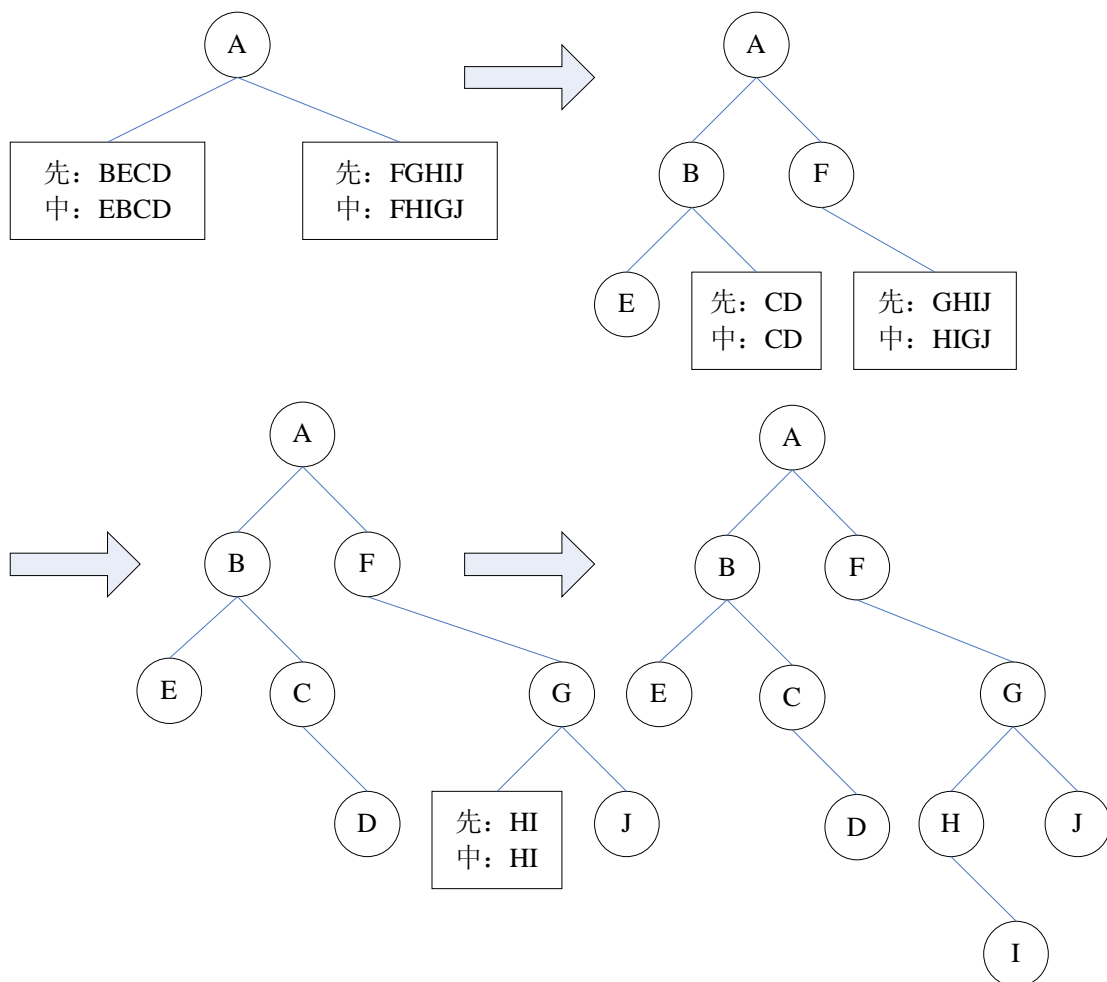
**解法：**

已知森林的先序遍历和中序遍历相当已知其对应的二叉树的先序遍历和中序遍历，  
即：

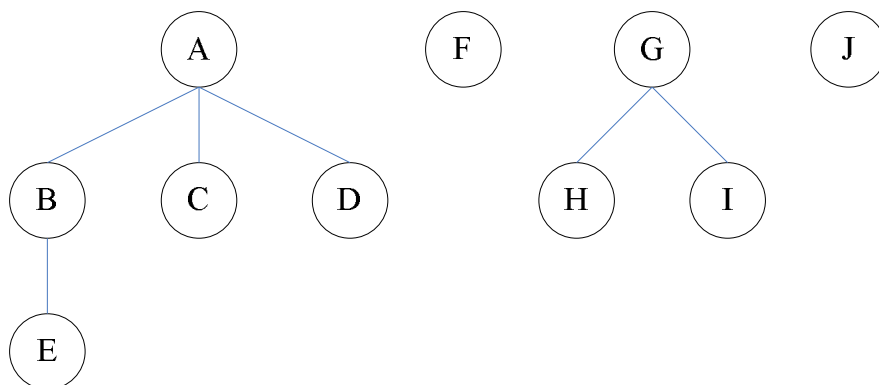
二叉树先序遍历序列：**ABECDFGHIJ**

二叉树中序遍历序列：**EBCDAFHIGJ**

则该二叉树构造如下：（由先序确定根结点，由中序确定左、右子树序列）

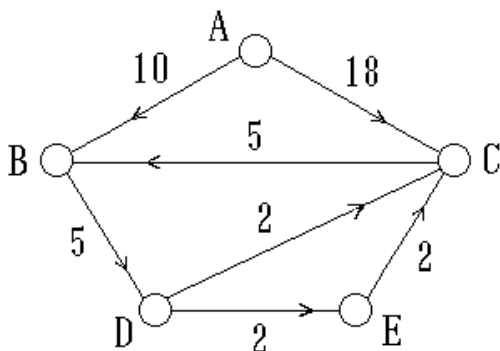


将该二叉树转换为对应的森林结构如下：



## 第 7 章 图

五、求解下面有向图的强连通分量，要求给出求解过程。



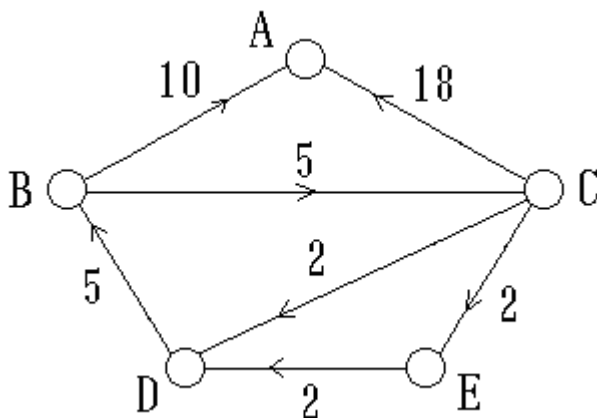
解答:

假定遍历过程以结点字母的先后为序。

以 A 为遍历结点做原图的深度优先遍历，按照其邻接点均遍历结束的顺序（即退栈顺序）所得序列 **finish** 为：

<b>finish</b>	<b>C</b>	<b>E</b>	<b>D</b>	<b>B</b>	<b>A</b>
---------------	----------	----------	----------	----------	----------

将原图各弧逆转，得到原图的逆转图如下：

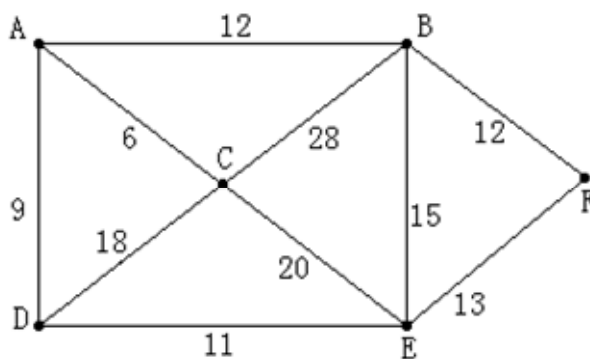


从 **finish** 序列的最后一个结点 A 出发在逆转图中做一次深度优先遍历，得到原图的一个强连通分量：{A}

再从 **finish** 序列剩余结点中的最后一个结点 B 出发在逆转图中做一次深度优先遍历，得到原图的一个强连通分量：{B, C, D, E}

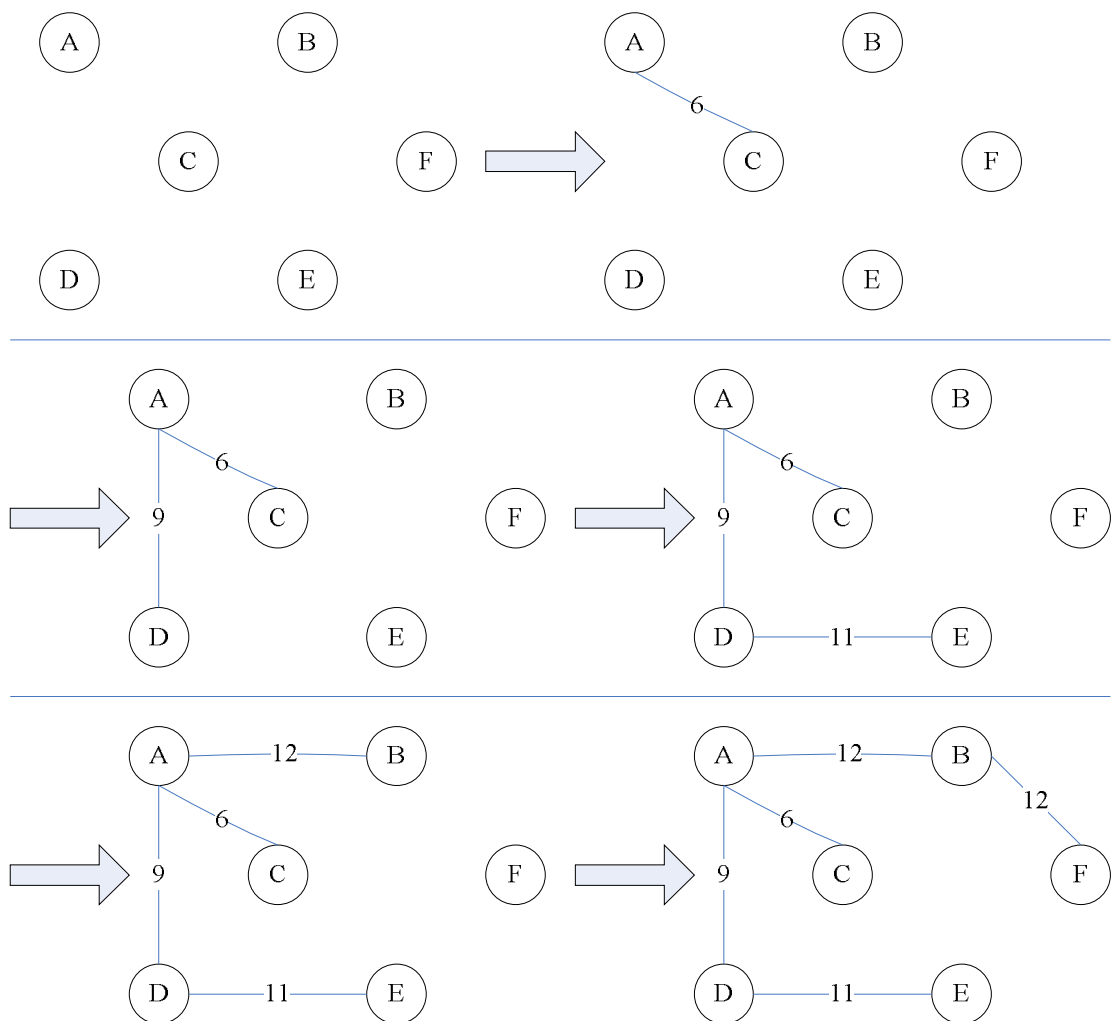
所以原图有两个强连通分量，分别是{A}和{B, C, D, E}。

六、已知图 G 如下所示，请求出该图的最小生成树，要求写明所采用的算法和中间过程。



解法一：采用普里姆算法（Prim）；以 A 作为算法的起始点。

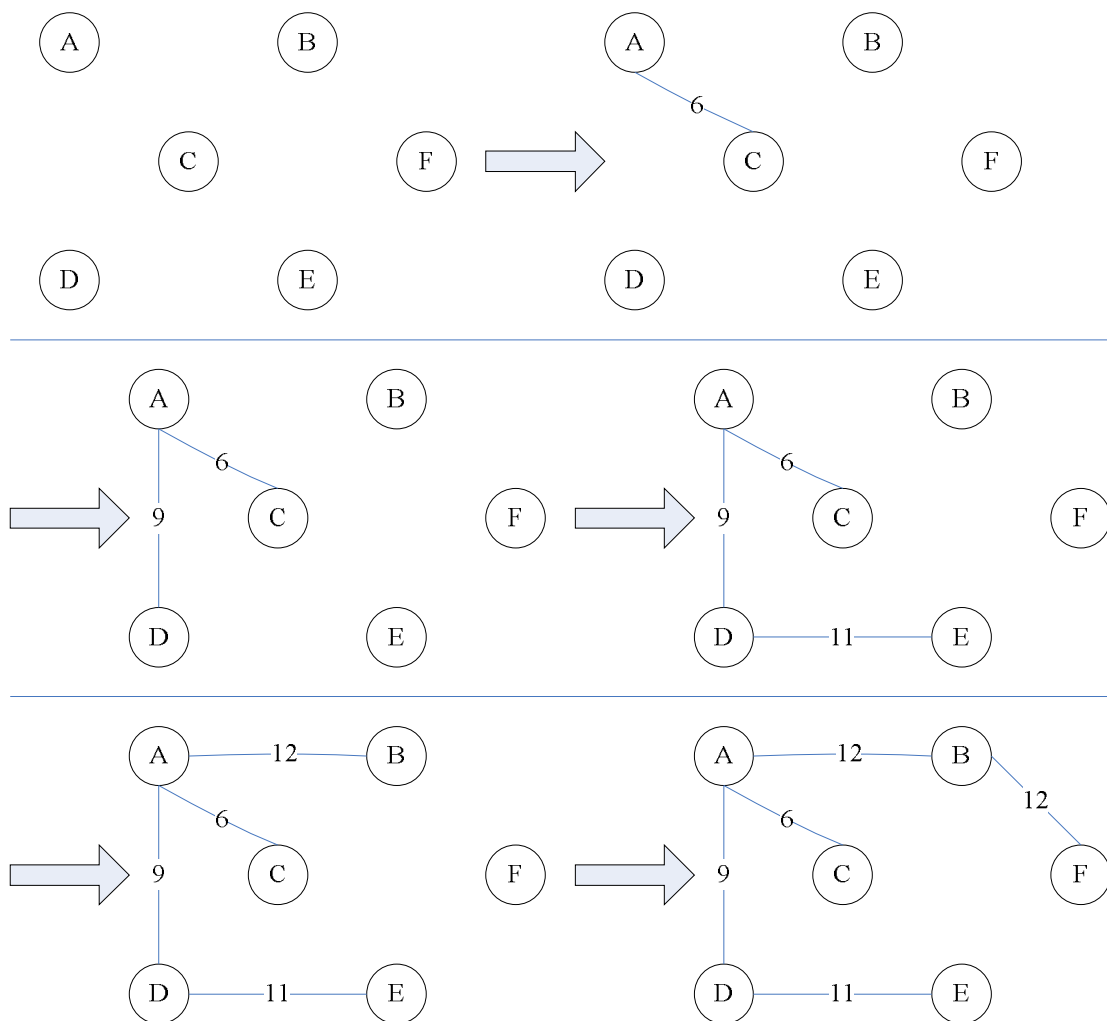
步骤	顶点 i closedge	2 B	3 C	4 D	5 E	6 F	U	V-U	k
0	Adjvex Lowcost	A 12	A 6	A 9			{A}	{B,C,D,E,F}	3 C
1	Adjvex Lowcost	A 12	0	A 9	C 20		{A,C}	{B,D,E,F}	4 D
2	Adjvex Lowcost	A 12	0	0	D 11		{A,C,D}	{B,E,F}	5 E
3	Adjvex Lowcost	A 12	0	0	0	E 13	{A,C,D,E}	{B,F}	2 B
4	Adjvex Lowcost	0	0	0	0	B 12	{A,B,C,D,E}	{F}	6 F
5	Adjvex Lowcost	0	0	0	0	0	{A,B,C,D,E,F}	{}	



解法二：采用克鲁斯卡尔算法 (Kruskal)

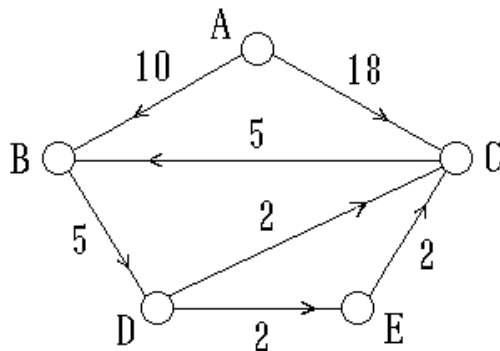
	VS	VT	Weight	选择标志 flag
0	A	B	12	1
1	A	C	6	1
2	A	D	9	1
3	B	C	28	0
4	B	E	15	0
5	B	F	12	1
6	C	D	18	0
7	C	E	20	0
8	D	E	11	1
9	E	F	13	0





步骤 顶点 Data	0	1	2	3	4	5
	隶属集合	隶属集合	隶属集合	隶属集合	隶属集合	隶属集合
A	1	1	1	1	1	1
B	2	2	2	2	1	1
C	3	1	1	1	1	1
D	4	4	1	1	1	1
E	5	5	5	1	1	1
F	6	6	6	6	6	1
选择边	AC	AD	DE	AB	BF	

七、已知图 G 如下所示，请按 Dijkstra 算法计算从顶点 A 到其它各个顶点的最短路径和最短路径长度，请给出求解过程。



解答:

终点	从 A 到其余顶点的 D 值和最短路径的求解过程			
	步骤: i=1	步骤: i=2	步骤: i=3	步骤: i=4
B	10 AB			
C	18 AC		17 ABDC	
D	$\infty$	15 ABD		
E	$\infty$		17 ABDE	
选择顶点 V	B	D	C	E
最短路径 P	AB	ABD	ABDC	ABDE

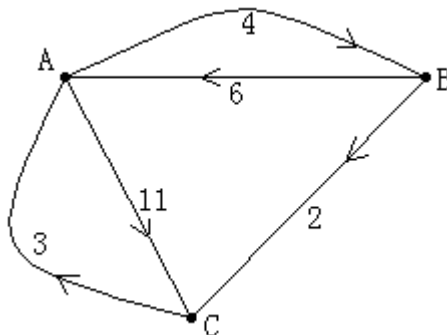
所以, A 到 B 最短路径为 AB, 最短路径长度为 10;

A 到 C 最短路径为 ABDC, 最短路径长度为 17;

A 到 D 最短路径为 ABD, 最短路径长度为 15;

A 到 E 最短路径为 ABDE, 最短路径长度为 17;

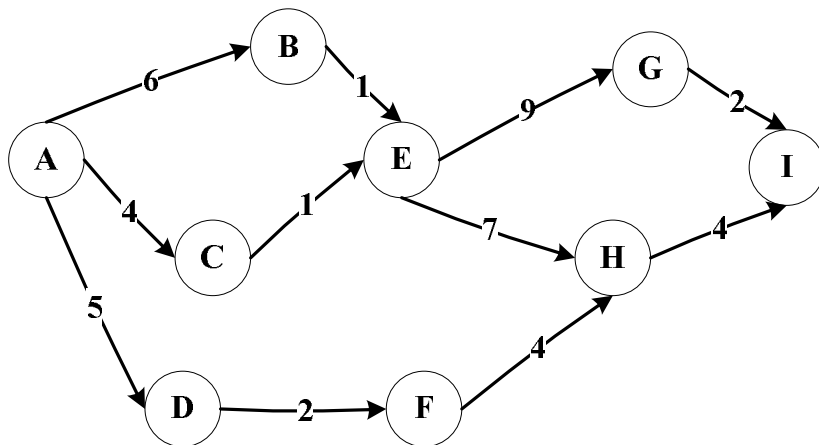
八、请用 Floyd 算法求出下图中任意一对结点之间的最短路径和最短路径的长度。



解答过程:

D		1A	2B	3C	P		1A	2B	3C
D0	1A	0	4	11	P0	1A		AB	AC
	2B	6	0	2		2B	BA		BC
	3C	3	$\infty$	0		3C	CA		
D1	1A	0	4	11	P2	1A		AB	AC
	2B	6	0	2		2B	BA		BC
	3C	3	7	0		3C	CA	CAB	
D2	1A	0	4	6	P2	1A		AB	ABC
	2B	6	0	2		2B	BA		BC
	3C	3	7	0		3C	CA	CAB	
D3	1A	0	4	6	P3	1A		AB	ABC
	2B	5	0	2		2B	BCA		BC
	3C	3	7	0		3C	CA	CAB	

九、已知 AOE 网如下所示，请求出该工程关键活动和关键路径，要求给出计算过程。



解答过程:

1、各顶点/事件的最早、最迟开始时间

顶点 时间	A	B	C	D	E	F	G	H	I
E 最早	0	6	4	5	7	7	16	14	18
L 最迟	0	6	6	8	7	10	16	14	18

2、各活动的最早、最迟开始时间

活动 时间	AB	AC	AD	BE	CE	DF	EG	EH	FH	GI	HI
E 最早	0	0	0	6	4	5	7	7	7	16	14
L 最迟	0	2	3	6	6	8	7	7	10	16	14

L-E	0	2	3	0	2	3	0	0	3	0	0
-----	---	---	---	---	---	---	---	---	---	---	---

所以该工程的关键活动为 AB、BE、EG、EH、GI、HI；有这些关键活动组成的关键路径为：ABEGI、ABEHI。

## 第九章 查找

十、假定一个线性表为  $L = (53, 25, 76, 20, 48, 14, 60, 84, 73, 36)$  进行散列存储，采用的 Hash 函数为  $H(K) = K \bmod 11$ ，当发生冲突时用线性探测法处理冲突，设 Hash 表的表长为 12，试构造 Hash 表，要求给出构造过程。并求出查找成功时的平均查找长度。

构造的哈希表如下：

0	1	2	3	4	5	6	7	8	9	10	11
36			25	48	14	60	84	73	53	76	20

哈希函数为  $H(k) = k \bmod 11$

线性再探测  $H_i(k) = (H(k) + d_i) \bmod 12$ ;  $d_i = 1, 2, 3 \dots$  当  $i = 1, 2, 3 \dots$

则哈希表构造过程如下：

$H(53) = 53 \bmod 11 = 9$

$H(25) = 25 \bmod 11 = 3$

$H(76) = 76 \bmod 11 = 10$

$H(20) = 20 \bmod 11 = 9$ (冲突)

$H_1(20) = (9+1) \bmod 12 = 10$ (冲突)

$H_2(20) = (9+2) \bmod 12 = 11$

$H(48) = 48 \bmod 11 = 4$

$H(14) = 14 \bmod 11 = 3$ (冲突)

$H_1(14) = (3+1) \bmod 12 = 4$ (冲突)

$H_2(14) = (3+2) \bmod 12 = 5$

$H(60) = 60 \bmod 11 = 5$ (冲突)

$H_1(60) = (5+1) \bmod 12 = 6$

$H(84) = 84 \bmod 11 = 7$

$H(73) = 73 \bmod 11 = 7$ (冲突)

$H_1(73) = (7+1) \bmod 12 = 8$ (冲突)

$H(36) = 36 \bmod 11 = 3$ (冲突)

$H_1(36) = (3+1) \bmod 12 = 4$ (冲突)

$H_2(36) = (3+2) \bmod 12 = 5$ (冲突)

$H_3(36) = (3+3) \bmod 12 = 6$ (冲突)

$H_4(36) = (3+4) \bmod 12 = 7$ (冲突)

$H_5(36) = (3+5) \bmod 12 = 8$ (冲突)

$H_6(36) = (3+6) \bmod 12 = 9$ (冲突)

$H_7(36) = (3+7) \bmod 12 = 10$ (冲突)

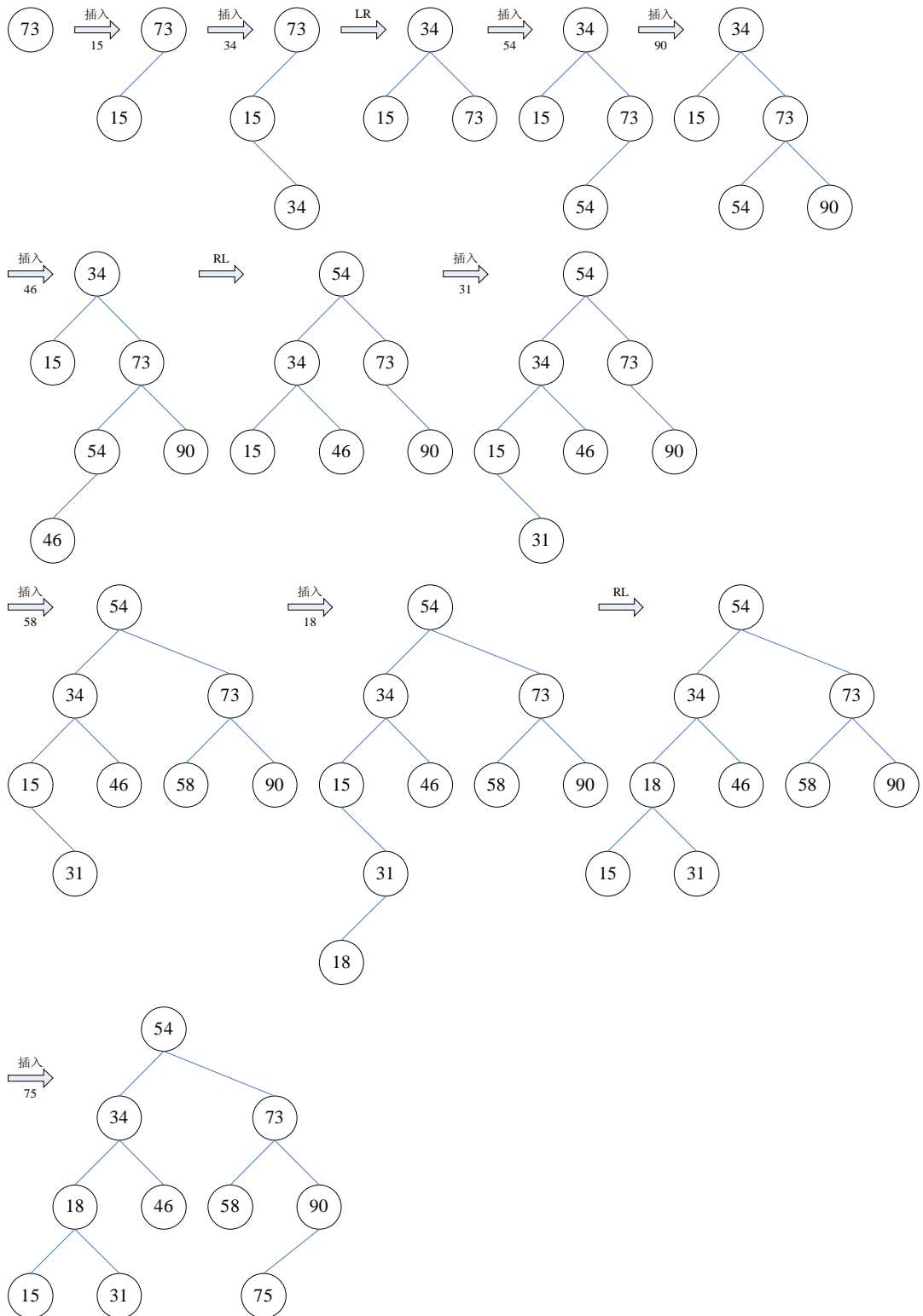
$$H8(36)=(3+8) \bmod 12=11 \text{ (冲突)}$$

$$H9(36)=(3+9) \bmod 12=0$$

查找成功时的平均查找长度为： $ASL=(1*5+2*2+3*2+10*1)/10=2.5$

十一、 输入关键字序列{73, 15, 34, 54, 90, 46, 31, 58, 18, 75 }， 建立一棵平衡二叉排序树，请给出构造过程。

平衡二叉排序树的构造过程如下所示：



## 第十章 内部排序

十二、 已知关键码序列{22, 55, 42, 48, 15, 30, 23, 94, 10}, 请用堆排序实现从小到大排序。要求写出排序过程。

**解答过程:** (用树的形式表示也可)

原始序列: {22, 55, 42, 48, 15, 30, 23, 94, 10}

初始建大顶堆: {94, 55, 42, 48, 15, 30, 23, 22, 10}

输出堆顶 94 (与堆中最后元素 10 交换), 重新建大顶堆: {55, 48, 42, 22, 15, 30, 23, 10}, 94

输出堆顶 55 (与堆中最后元素 10 交换), 重新建大顶堆: {48, 22, 42, 10, 15, 30, 23}, 55, 94

输出堆顶 48 (与堆中最后元素 23 交换), 重新建大顶堆: {42, 22, 30, 10, 15, 23}, 48, 55, 94

输出堆顶 42 (与堆中最后元素 23 交换), 重新建大顶堆: {30, 22, 23, 10, 15}, 42, 48, 55, 94

输出堆顶 30 (与堆中最后元素 15 交换), 重新建大顶堆: {23, 22, 15, 10}, 30, 42, 48, 55, 94

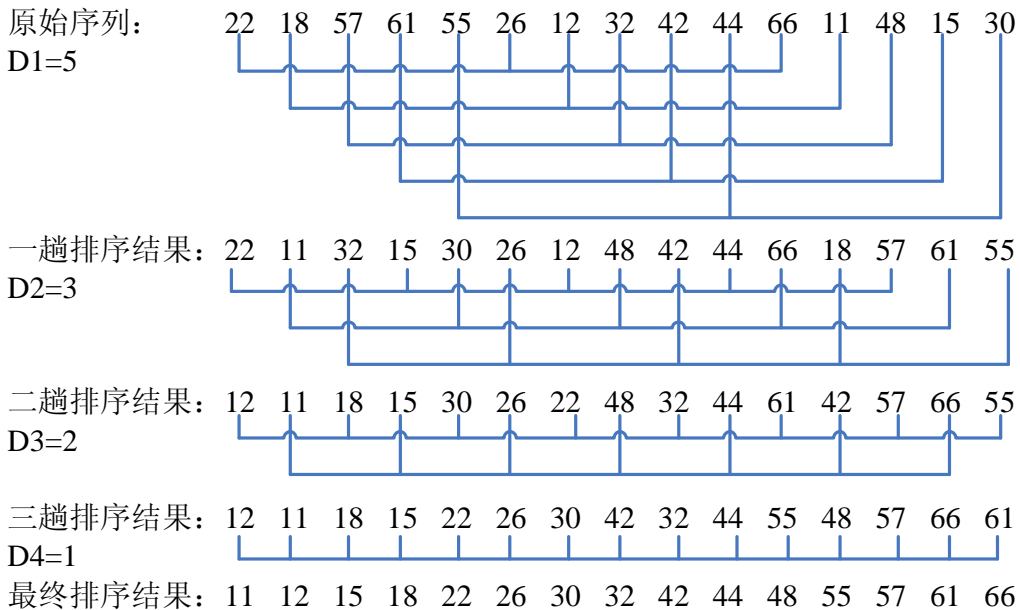
输出堆顶 23 (与堆中最后元素 10 交换), 重新建大顶堆: {22, 10, 15}, 23, 30, 42, 48, 55, 94

输出堆顶 22 (与堆中最后元素 15 交换), 重新建大顶堆: {15, 10}, 22, 23, 30, 42, 48, 55, 94

输出堆顶 15 (与堆中最后元素 10 交换), 得到最终排序序列: 10, 15, 22, 23, 30, 42, 48, 55, 94

十三、 已知关键码序列{22, 18, 57, 61, 55, 26, 12, 32, 42, 44, 66, 11, 48, 15, 30}, 请用希尔排序实现从小到大排序, 增量序列为  $d_1=5, d_{i+1}=(d_i+1)/2$ 。请给出排序过程。

**解答过程:**



十四、 已知关键码序列{49, 38, 65, 97, 76, 13, 27, 50}, 请用快速排序实现从小到大排序, 排序限定采用序列首元素作为枢轴。请给出排序过程。

解答过程:

原始序列: 49 38 65 97 76 13 27 50

枢轴=49, 一趟快排划分结果:

{27 38 13} **49** {76 97 65 50}

枢轴=27, {27 38 13}进行快排划分,

枢轴=76, {76 97 65 50}进行快排划分,

二趟分别快排划分结果:

**49**

{13} **27** {38} {50 65} **76** {97}

{13}、{38}、{97}仅有一个元素, 不用继续快排划分操作。

枢轴=50, {50 65}进行快排划分,

三趟分别快排划分结果:

**49**

**27**

**76**

{13} {38} **50** {65} {97}

最终排序结果: {13 27 38 49 50 65 76 97}

十五、 已知关键码序列{278, 109, 63, 930, 589, 184, 505, 269, 8, 83}, 请用基数排序实现从小到大排序。请给出排序过程。



**解答过程：**

原始序列： 278 109 063 930 589 184 505 269 008 083

以关键字的个位进行分配、收集的结果为：

930 063 083 184 505 278 008 109 589 269

以关键字的十位进行分配、收集的结果为：

505 008 109 930 063 269 278 083 184 589

以关键字的百位进行分配、收集的结果为：

008 063 083 109 184 269 278 505 589 930

最终排序结果：

{008 063 083 109 184 269 278 505 589 930}