

1850 PRINT ".....风量 风压 与 风量效率 曲线....."

1860 PRINT "H"; "Pa"; TAB(68); "SL"; TAB(71); "%"

1870 FOR I=100 TO 1000 STEP 100

1880 PRINT "I"; " "

1890 IF I/300 = 10 THEN PRINT TAB(32); "Q=10";

1900 IF I/300 = 8 THEN PRINT TAB(32); "Q=8";

1910 PRINT TAB(71); " "; I/30

1920 NEXT I

1930 PRINT TAB(2); 600; " "; TAB(71); " "; 20

1940 PRINT TAB(3); 0; TAB(72)

1950 LINE(560,137)-(560,40)

1960 LINE(48,40)-

1970 GOSUB 2030

1980 FOR I=400 TO 2800 STEP 400

1990 PRINT TAB(5+I/50); I;

2000 NEXT I

2010 PRINT "Q"; "m³/min"

2020 GOTO 2080

2030 LINE(48,137)-(560,137)

2040 FOR I=1 TO 8

2050 J=48+I*61

2060 LINE(J,137)-(J,134)

2070 NEXT I; RETURN

2080 FOR I=1 TO A3

2090 YMAX=0

2100 FOR J=1 TO D

2110 QQ(I,J)=Q1(J)*1000; HH(I,J)=HC1(J); IF YMAX<HC1(J) THEN

YMAX=HC1(J)

2120 NEXT J; GOSUB 2120

2130 FOR Q=XMIN TO XMAX STEP 1

2140 H=0

2150 FOR I=1 TO G-1; H=H+B(I)*Q*(I-1)

2160 NEXT I

2170 PSET(48+Q/400*61,137-H/600*18); NEXT Q

2180 FOR I=1 TO A3

2190 YMAX=0

2200 FOR J=1 TO D

2210 QQ(I,J)=Q1(J)*1000; HH(I,J)=SL1(J); IF YMAX<SL1(J) THEN YMAX=SL1(J)

矿井通风微机程序 设计与应用

主编 沈斐敏

编著 沈斐敏 刘绍权 陈文礼

煤炭工业出版社

矿井通风微机程序设计与应用

主编 沈斐敏
编著 沈斐敏 刘绍权 陈文礼
主审 范明训
审稿 范明训 夏福志

煤炭工业出版社

(京)新登字042号

图书在版编目(CIP)数据

矿井通风微机程序设计与应用/沈斐敏主编;沈斐敏等
编著. - 北京:煤炭工业出版社, 1995. 7

ISBN 7-5020-1138-2

I. 矿… II. ①沈…②沈… III. 矿井通风-程序设计-
微型计算机 IV. TD72

中国版本图书馆CIP数据核字(95)第02376号

矿井通风微机程序设计与应用

主编 沈斐敏

责任编辑:辛广龙

*

煤炭工业出版社 出版

(北京安定门外和平里北街21号)

煤炭工业出版社印刷厂 印刷

新华书店北京发行所 发行

*

开本787×1092mm¹/16 印张18³/4插页1

字数444千字 印数1—1,500

1995年8月第1版 1995年8月第1次印刷

书号 3906 G0317 定价18.60元

前 言

60年代以来,电子计算机技术发展极为迅速,特别是1983年以来,微型机技术的崛起和发展,使计算机在矿井通风中得以普及使用成为可能。作者多年来在进行科研教学的同时,与现场工程技术人员结合,广泛地把微机应用于矿井通风实际,有效地解决了许多生产实际问题。大量的实践表明,微型电子计算机的应用给矿井通风工作提供了强有力的工具。它的使用,大大地加强与改善了通风工作,提高了矿井通风的科学水平,促进了矿井通风的科学管理。

随着生产和科学技术的发展,掌握这一现代技术,对现在和今后从事矿山通风管理人员均有深远的意义。

作者期望,本书的出版能有助于广大矿山通风工作者学习和掌握微机技术,使其能更广泛地应用于科研、教学和生产实际。

本书是根据作者在1984年3月编写的“微型电子计算机在矿井通风中应用”讲义,并经多次修改补充和福州大学12届采矿专业本科生及多次培训班教学使用后,于1992年12月所形成的“微机在矿井通风中应用”教材的基础上,又吸收了近2年来的研究与应用成果写成的。

本专著主要讨论了微机在矿井通风中应用的有关问题。书中对程序所依据的计算模型、编程方法和编程步骤进行了较为详细的阐述,其目的是使读者对书中所讨论的问题的程序设计思想、设计方法及技巧能较容易理解与借鉴,以提高自行编程能力和优化程序。

书中对所讨论的问题均附有电算源程序和计算例题,且对各程序的使用做了详细的说明。

本书中主要的程序均以BASIC语言的形式出现以方便读者学习和使用;书中还特别注意数据管理功能较强的数据库软件(如汉字FOXBASE和dBASE III)与BASIC语言的配合使用,使计算与数据处理得到了充分的体现,有效地提高了微机在矿井通风中的管理功能。书中所有的BASIC程序稍经修改,均可在Turbo和Ture Basic语言中运行。

本书在编写过程中,特别得到了山东矿业学院范明训教授的具体指导,西安矿业学院刘冠姝教授也给予了诸多指导;此外,尚得到福州大学、福建省煤炭总公司、龙岩矿务局、永定矿务局、上京矿务局、苏邦煤矿、天湖山矿务局和邵武煤矿等单位的有关领导和通风工程技术人员的大力支持和帮助,在此,谨向上述有关单位和人员致以诚挚的感谢。

由于作者的水平所限,书中定有疏漏之处,敬请读者批评指正。

作 者

1994年10月

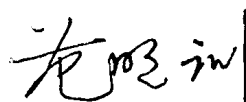
序

矿井通风是矿山开采的一个重要环节，是矿井实现安全生产的保障。长期以来，从事矿井通风工作的专业人员一直在致力于矿井通风技术及管理水平的提高。随着计算机技术的发展，如何实现计算机技术在矿井通风中的广泛而有效的应用，从而提高矿井通风技术及管理水平，已成为目前矿井通风工作的热点。

为了推广计算机技术的应用、提高应用水平，沈斐敏副教授根据多年来从事矿井通风与安全方面的教学及科研工作所积累的成果与经验，撰写了这本专著。本书的主要特点是理论联系实际，其重点阐述了如何应用计算机技术解决矿井通风工作中常见的问题，还突出介绍了程序编写的设计思想、方法及技巧，充分体现了高级语言与数据管理系统的有机结合。

本书材料充实、内容新颖、条理清楚、方法与应用兼备、程序与实例具有、涉及面广、有一定的深度和较强的实用性。书中的井巷摩擦风阻的计算、最大（小）阻力线路的选择及有关数据传递处理等内容中的构想与计算方法均是作者较早提出的。可见，作者在撰写本书过程中是精心构思、独具匠心的。

可以肯定，本书的出版对广大通风工作者学习及掌握计算机技术、实现矿井通风工作科学化和管理现代化，必将起到重要的促进作用，也将为我国矿井现代化建设做出新的贡献。



1995年5月

目 录

第一章 程序设计初步	1
第一节 编程概述	1
第二节 求算水蒸汽绝对压力 p_{sa} 值程序的编写	11
第三节 空气密度解算程序的编写	15
第四节 对角网络解算程序的编写	18
第五节 通风机特性曲线模拟程序的编写	23
第二章 矿井通风网络解析	31
第一节 几个基本术语	31
第二节 通风网络中风流流动的基本规律	31
第三节 风量平衡定律程序的编写	34
第四节 风压平衡定律程序的编写	38
第五节 通风网络的解算	40
第六节 自然分风程序的编写	42
第七节 网孔的选择	52
第八节 最小树的选择算法	56
第九节 加边法选择最小树的程序的编写	58
第十节 破圈法选择最小树的程序的编写	62
第十一节 矿井复杂通风网络解算程序的编写	67
第三章 矿井通风阻力测定	89
第一节 通风阻力测定数据处理计算式	89
第二节 程序设计框图	91
第三节 压差计法的程序编写	91
第四节 基点法的程序编写	105
第五节 压差计法与气压计测定法的共用程序的编写	113
第四章 矿井主通风机性能鉴定	138
第一节 局部通风机性能测定参数的计算及性能曲线绘制程序的编写	138
第二节 矿井主通风机性能测定	154
第五章 矿井主通风机的选择	169
第一节 主通风机选择程序的设计思想	169
第二节 矿井主通风机选择程序	172
第三节 程序应用举例	185
第六章 掘进通风与风量调节	199
第一节 选择计算掘进通风有关参数的程序	199
第二节 风量调节	207
第七章 矿井通风设计	213
第一节 采区所需风量的计算	213

第二节	矿井总风量及风量分配计算	221
第三节	井巷风阻计算程序	224
第四节	矿井通风阻力计算程序	237
第八章	应用网络解算程序进行矿井通风系统改造实例	257
第一节	矿井现状与通风存在的问题	257
第二节	矿井通风系统改造方案的提出	258
第三节	两方案的通风前期网络验算及分析	259
第四节	两方案的通风后期网络验算及分析	278
第五节	方案比较与选定	286
第六节	程序编写及使用的若干注意点	289
参考文献		292

第一章 程序设计初步

第一节 编程概述

计算机是一种功能很强的机器，但它仅是一种机器，其本身不能代替人类的思维，它只有在接受了由人类思维所转换的信息后，才能进行超人类的计算速度进行运算解题，所以，给予计算机以能够接受的信息是保证计算机运行的一个关键。人们通常把这种信息的传递媒介称之为程序。从实质上说，程序是人类解决问题的思维反映，它是用人类和计算机二者都能认识的语言进行编写的。

编写程序是利用计算机来解决问题的一个重要步骤，它要求人们对所需要解决的问题进行周密的考虑，确定出解决问题所牵涉到的内容，选择出一种能够解决问题的方法，只有做完这些工作后才能着手用语言来描写。

一般说来，编写一个程序应做以下几个工作：

- (1) 确定解决问题的基础（数学）模型；
- (2) 确定解决问题的思路（常用流程框图来表示）；
- (3) 根据解题思路，分段进行具体的处理（选择恰当的计算方法，也可划出各段的细框图）；
- (4) 按照BASIC的语言法则，把整个过程用不同语句进行组合，即构成程序。

具体说来是：

(1) 确定解决问题的基础或数学模型。这一步是编写程序的依据，我们要把所要解决的问题抽象成一个数学或基础模型，通过对这种模型的解算达到解题的目的；

(2) 确定解决问题的思路，它指的是，当第一个问题提出后，我们应沿着哪条路子才能把这个模型解出来。整个解题思路要明确，究竟要分几步走，要心中有数，这一步，通常是用粗框表示出来；

(3) 当粗框形成以后，要确定各个步骤的详细处理步骤及计算方法。确定完后，可用细框图来表示，当然不用也行。为了说明问题起见，本书划出细框图；

(4) 当上面工作做完后，用不同的语句描述、组合，就构成了程序。

一、例题分析

为加深对上内容的理解，下面以一例说明。

在矿井通风设计中，其中一个主要任务就是找出困难通风时期的通风路线，并求出该线路的通风阻力（最大阻力），以做为选择风机的依据。在传统处理方法中，最大线路的选择是靠人为经验决定的，这样容易产生误差；较为精确的一种就是先算出网络中各支路的风压值，然后把各线路的风压值大小进行比较，择出最大的一条，其值也就是所求的最大阻力值。下面就对此问题如何处理进行分析。

1. 确定解决问题的基础模型

大家知道，不同的通风网络结构不一样，有的简单，有的复杂，所以，在建立基础模

型时,应寻找出各种类型网络的共性,根据这种共性,利用最简单的网络做为基础模型,从解决这一基础模型入手,编写出一个能适用于任何形式的同类问题的通用程序。

在编写该程序时,依据网络所具有的共性,以一个简单的网络做为编程基础,这个网络如图1-1所示:

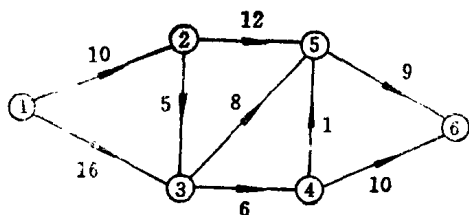


图 1-1 基础网络图

我们从这个基础网络图上可以看出,该图是点、边的集合。

网络中的圆圈代表图中的点,通常称之为结点或节点(圆圈中的数字是对各结点的编号);二个结点之间包着一边,这条边若用图论的术语来表达,则称之为“线”,各边旁边写着的数字称之为“权”,权值的意义可以人为定义。对于此例,它的意义指的是

每条边的风压值;另外,图上还有箭号,图论中指的是边的指向,此例的特定含义是风流的方向。我们把箭头所指的结点叫做下结点,常用 j 来表示,箭头反方向所指的结点称为上结点,常用 i 来表示。

另外,我们把网络中的第一个编号称为起始节点号,用 s 这个符号来表示。把网络中最后一个编号称为终止节点号,用 z 来表示。对于图1-1所示网络,起始节点 s 即是节点①,终止节点 z 即为节点⑥。

从该网络中寻找出始终二点间的最长线路,即为通风阻力最大线路。

例如,对于上图,通过手算,我们可以知道始、终二点各线路的值,即:

- ①—②—⑤—⑥线路的值为31;
- ①—②—③—⑤—⑥线路的值为32;
- ①—②—③—④—⑥线路的值为31;
- ①—③—⑤—⑥线路的值为33;
- ①—③—④—⑥线路的值为32。

把上述各条线路进行比较后,要求的最大线路就是①—③—⑤—⑥,其值33为通风阻力最大值。

通过基础模型的分析知道,要求解出阻力最大线路,则先要累加各线路中各边权值,然后进行比较。手算可采用上面的方法,那么对于电子计算机,又要怎么办呢?该走哪条路子才能达到选出最大线路的目的呢?采用图论的理论,可对图中的各行程链的大小进行统计比较,利用数组链表处理的方法选择出最大的线路。

2. 流程框图(解题思路)

流程框图如图1-2所示。

3. 根据解题思路,分段进行具体处理

- (1) 输入数据;
- (2) 构成链表 A 的框架。

为了能更好的说明问题,讲这个问题以前,先引进联组的概念。

上节点和直接与其相联的所有下节点所构成的一个集合,则称联组。联组由上节点号来命名。

例如图2-1中的点①与②、③构成一个集合，我们就把由这三点构成的集合叫做联组，并把①结点做为这个联组的名称，这样，这个联组就叫第①联组。

要构成链表A框架，需用这样的程序段来实现，其细框如图1-3所示：

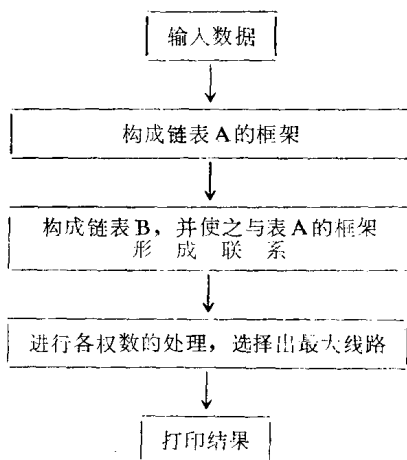


图 1-2 流程框图

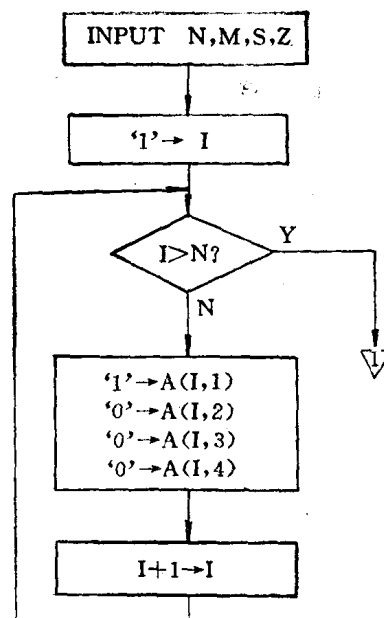


图 1-3 细框图

从图1-3细框可以看出，要构成表A的框架，首先要输入4个数据，其中：

- N 代表网络中结点的总数6；
- M 代表网络各边支路的总数9；
- S 起始节点号1；
- Z 终点号6。

输入这4个数据后，由程序段执行完毕后，则可构造出链表A。

这一个程序段实际是一个循环，I为循环变量，1为循环变量初值，N为循环变量终值。

由于N所赋的值是6，所以这个循环要执行6次，其运行情况如下示：

当I=1时	当I=2时	当I=3时	当I=4时	当I=5时	当I=6时
A(1,1)=1	A(2,1)=1	A(3,1)=1	A(4,1)=1	A(5,1)=1	A(6,1)=1
A(1,2)=0	A(2,2)=0	A(3,2)=0	A(4,2)=0	A(5,2)=0	A(6,2)=0
A(1,3)=0	A(2,3)=0	A(3,3)=0	A(4,3)=0	A(5,3)=0	A(6,3)=0
A(1,4)=0	A(2,4)=0	A(3,4)=0	A(4,4)=0	A(5,4)=0	A(6,4)=0

从运行的结果看，当执行完6次循环以后，它则构成了这样的一个框架。

	1)	2)	3)	4)
A(1,	1	0	0	0
A(2,	1	0	0	0
A(3,	1	0	0	0
A(4,	1	0	0	0
A(5,	1	0	0	0
A(6,	1	0	0	0

代表↑ 第1格↑ 第2格↑ 第3格↑ 第4格↑
 结点号码 标志 从起点到本结点的 最长行程链中与本结 本联组的联结首址
 最长行程 点相邻的上结点号

此表是由二维数组组成的。对于每个下标，我们都赋予它一个特定的含义，对于行标(1, 2, 3, 4, 5, 6)来说，它代表的是一项，这一项用来表征对应号码的节点，或者说，数组中的行标变量是用来代表要对应的结点号的；而对于列标变量这一项(1), 2), 3), 4))，每一个列号表示其中的一格，而各格都给于一个特定的涵义：

第1格做为标志格，当它为“1”时，表示这一项待查，如果计算机查过此项后，就把它置“0”；

第2格表示从起点到本结点的最长行程，用来装放风压。开始时，设置一个足够小的初值，供比较选用；

第3格表示最长行程链中与行标所示的结点相邻的上结点号；

第4格表示本联组（联组的名称就是行标中的结点号）联结的第一个址码。

从上表可以看出，利用这个表，就可以根据不同的下标在表内填进或者取出数据。

比如结点①—②之间这条边，在A(2,)行中，在第4格可以填入1、在第2格可以填入10、在第3格可以填入1，但仅有此数组还不能完全解决问题，尚需构造出一个链表B，用它和表A的框架沟通起来一起使用。建立链表B的目的是利用链表B来存放联组中下结点的数据，与表A框架中第4格联系起来。由表A中的第4格提供联接的首址，根据它就可以查出表B中有关联组的情况。

(3) 构成链表B与表A框架的联系。

这一工作可用如图1-4所示的程序段来实现。

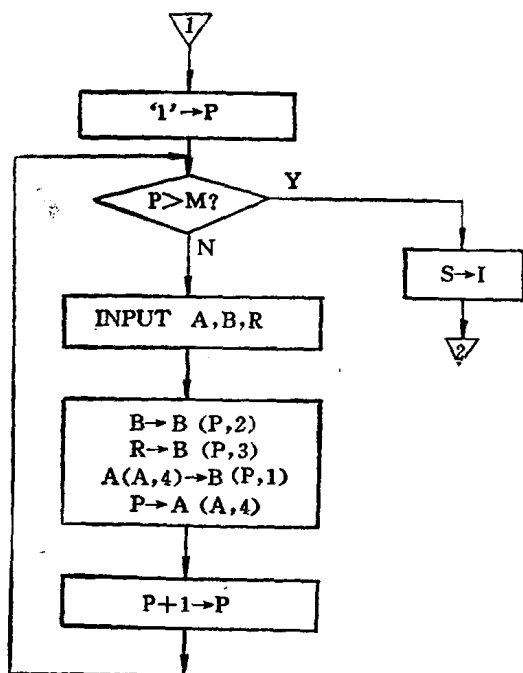


图 1-4

当采用上述程序段构造出表A框架后,则要求从键盘上输入各边的数据。这些数据包括各边的上节点A、下节点B和各边的风压值R,这些值的输入顺序可以是任意的,但必须包括全部结点,且包括各边的风压值。各条边上的风压值不允许重复输入。

假定从键盘上输入的数据按以下的顺序进行:

上结点	下结点	边 (各支路的阻力)
A	B	R
1	2	10
1	3	16
2	3	5
2	5	12
3	4	6
3	5	8
5	6	9
4	6	10
4	5	1

那么,这个程序段的运行情况是:

P = 1 时			
A = 1, B = 2, R = 10			
B (1, 2) = 2			
B (1, 3) = 10			
∴ A (1, 4) = 0			
∴ B (1, 1) = 0			
∴ A (1, 4) = 1			

P = 2 时			
A = 1, B = 3, R = 16			
B (2, 2) = 3			
B (2, 3) = 16			
∴ A (1, 4) = 1			
∴ B (2, 1) = 1			
∴ A (1, 4) = 2			

通过上述的二次循环,就把联组 I (在A数组中的行标所示,它为联组的上结点,也是命名点)的情况全部登记在表A框架中的第一项第4格和表B的第一、第二项中。

这样反复地运行下去,执行完9次循环后,则可构造出表B,并形成表B与表A框架的联系形式:

	1)	2)	3)	4)
A (1,	1	0	0	0, 1, 2
A (2,	1	0	0	0, 3, 4
A (3,	1	0	0	0, 5, 6
A (4,	1	0	0	0, 8, 9
A (5,	1	0	0	0, 7
A (6,	1	0	0	0

↑ 第4格

	1)	2)	3)
B (1,	0	2	10
B (2,	1	3	16
B (3,	0	3	5
B (4,	3	5	12
B (5,	0	4	6
B (6,	5	5	8
B (7,	0	6	9
B (8,	0	6	10
B (9,	8	5	1

↑ 联址 ↑ 下结点号 ↑ 风压

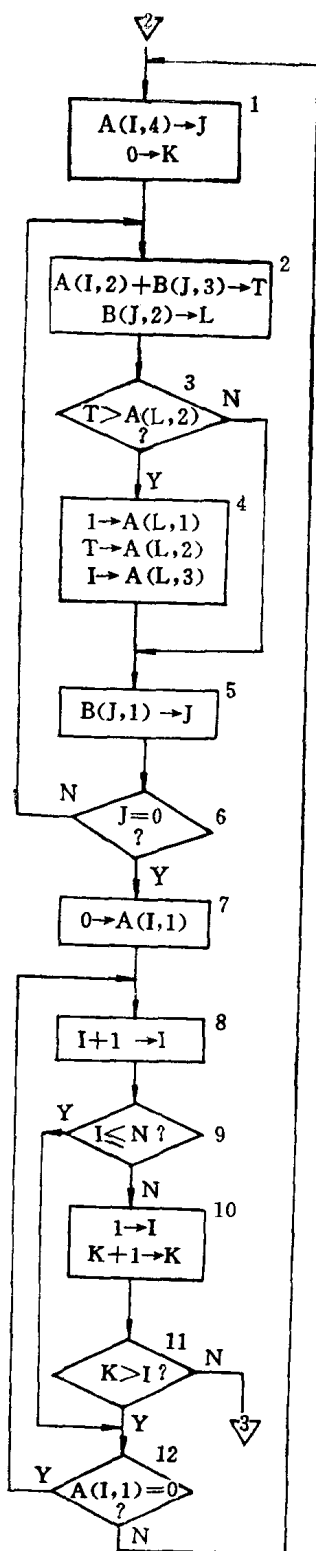


图 1-5

看表 B ，它也是由 2 维数组所构成的。下标变量中的行号代表一项，用来代表一条边，因为网络有 9 条边，所以行标有 9 个；下标变量中的列号表示一格，每一格赋于一特定的涵义：

第 1 格：联接的地址；

第 2 格：相应边的下结点号；

第 3 格：各边的风压。

通过这样处理后，网络中各联址的情况就全部登记在表 A 与表 B 中，计算机只要在表 A 的第 4 格中查明联接首址号，则马上可以在表 B 中找出该联组要对应的下节点和各边的风压值。

如第 1 联组的情况：

计算机若要查第一联组的情况，它首先在 $A(1, 4)$ 格中查明它的值是“2”，那么它就知道该联组的第 1 个下节点号是在表 B 中的第二项中的第 2 格里，其标号为“3”，且①—③之间边的风压为 $B(2, 3)$ 中的数值，其值为 16。

此值查完后，它接着要查问是否还有第 2 个下结点，此时，它查询 $B(2, 1)$ 中的第一格， $B(2, 1)$ 的值为“1”，则表明，该联组还有一个下结点号，它的位置是在表 B 的第二项中；在 $B(1, 2)$ 中查询出标号为“2”，则表明联组“1”的另一个下结点号是 2，则在 $B(1, 3)$ 中查了它所对应的风压值为 10。

查完后，它又在 $B(1, 1)$ 中查询看 1 联组是否还有其它下结点，由于 $B(1, 1)$ 为“0”，则说明 1 联组中的下结点都已查完。接到这个指令后，它就转向查其它联组的情况。这样反复进行，直至查完全部联组的情况，才转向执行其它的工作。

上面我们详细地讨论了联组 1 的情况，至于其它联组也是这样查寻的：

上面就是形成表 B 及表 B 与表 A 框架联系的情况。通过这样处理，计算机查找联组的速度就大大的加快。

当上述工作做完后，接着就应进行链权的处理工作。

(4) 进行各链权的处理，选择出最大线路。

这里的链权处理，指的是对各边的风压值进行累加、比较，然后选择出最长的线路。

这一步工作可用图 1-5 所示的程序段来执行。

各细框的作用是：

框 1：取出联组 I 中的联结首址，并把计数器 K 的初值赋为“0”；

框2: 起始节点至 I 节点的风压 $A(1, 2)$ 加上节点 I 至当前下节点 (I 所联接的下结点) 的风压 $B(J, 3)$, 把其结果放入 T , 并取出当前下结点号放入 L ;

框3: 把迄今为止从起始点至结点 L 的时间与 T 相比较;

框4: 若 T 大, 则将结点 L 的标志置 “1”, 以便进一步查访, 同时将 T 值代入 $A(L, 2)$, 并将结点 I 作为 L 的有效上结点, 建立新的最长行程链;

框5: 取联组 I 的另一个联址 (以便寻找另一个下结点);

框6: 是否链尾? 不是则转框 2 (表示是否都找完了联组的所有下结点, 若没找完, 则转 2。若找完, 则 7);

框7: 若是链尾, 则将结点 I 中的标志置 “0”, 表示联组 I 查访完毕;

框8: 取下一结点号;

框9: 超最大结点号否? 若未超过, 转框12;

框10: 超过最大结点号, 从结点 1 开始, 计数器 K 加 1;

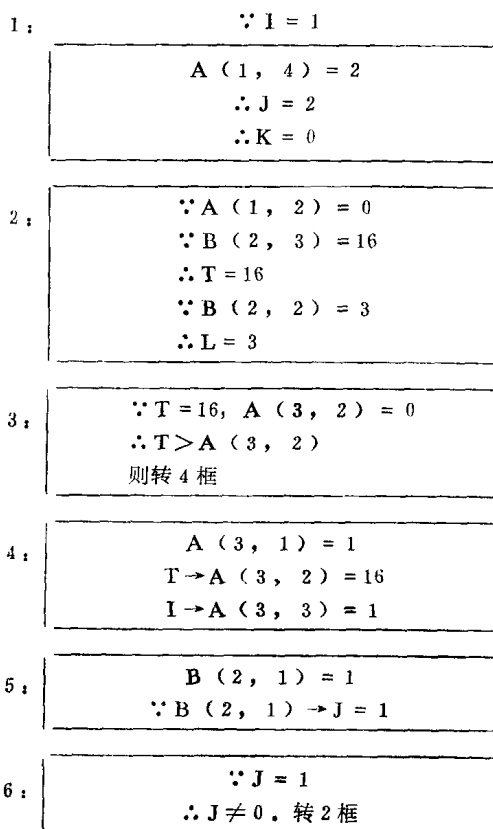
框11: 全部联组都处理完否?

框12: 联组 I 是否还需查访? (标志若为 “1”, 表示需要查访, 转 1 框, 否则转 8 框)。

上面是该程序段中各个细框的作用, 下面, 看看它的运行情况。

在执行完图1-5所示的程序段以后, 则把网络的起始点号 $S \rightarrow I$, $\because S = 1$, 要执行这一框开始时, I 全部为 1。

第一框的执行情况是:



2'	$A(1, 2) = 0$ $B(1, 3) = 10$ $\therefore T = 10$ $\because B(1, 2) = 2$ $\therefore L = 2$
3'	$\because T = 10, A(2, 2) = 0$ $\therefore T > A(2, 2)$ 则转 4 框
4'	$A(2, 1) = 1$ $A(2, 2) = 10$ $A(2, 3) = 1$
5'	$B(1, 1) = 0$ $\therefore J = 0$
6'	$J = 0$ $\therefore$ 转 7 框
7	$0 \rightarrow A(I, 1)$ $\therefore A(1, 1) = 0$
8	$1 + 1 \rightarrow I, I = 2$
9	$I (= 2) < N (= 6)$ 则转 12 框
12	$\because A(2, 1) = 1$ $\therefore A(2, 1) \neq 0$ 转 1 框, 开始查第 2 联组

到此, 就是查访一联组的全部运行过程。我们分析一下运行后的情况可以发现, 在查完 1 联组后, 表 A 变成以下形式:

	1)	2)	3)	4)
$A(1,$	0	0	0	2
$A(2,$	1	10	1	4
$A(3,$	1	16	1	6
$A(4,$	1	0	0	9
$A(5,$	1	0	0	7
$A(6,$	1	0	0	0

这样, 从联组 1 开始, 一直查下去, 直到查完全部的联址为止。
查完后表 A 中的各值如下所示:

	1)	2)	3)	4)
A(1,	0	0	0	2
A(2,	0	10	0	4
A(3,	0	16	1	6
A(4,	0	22	0	9
A(5,	0	24	3	7
A(6,	1	33	5	0

从上表就可以直观的看出，最长线路是①—③—⑤—⑥，其值是33。

上面详细地讨论的各个链权处理、选择最大线路的整个过程与步骤是：从起点开始，依次递增结点的号码，访遍所有的结点。查访结点 I 时，先看标志，若 $A(I, 1)$ 为 0，表示勿需再查；若 $A(I, 1)$ 为 1，就进行联组 I 的处理，处理时，利用 $A(I, 4)$ 中的 联结首址 J 查访所有的下结点，对于联组中的每一个下结点是从 $B(J, 2)$ 取出的放入 L 的那个结点数；再取出从结点 I 到结点 L 的风压同 $A(I, 2)$ 中的最大风压值相加，结果放入 T ；取出从开始结点经过另一途径到结点 L 的最大风压值 $A(L, 2)$ 同 T 相比，若 $T > A(L, 2)$ ，就表示找到了一条更大的通向 L 的途径，于是将 $A(L, 1)$ 的标志置 1，以便在后来进一步沿这条途径查访下去，又将 T 填到 $A(L, 2)$ ，对最大风压加以更新；再将结点 I 填入 $A(L, 3)$ ，表示迄今为止从开始结点到结点 L 的最长行程上 I 是 L 的上结点，若 $T < A(L, 2)$ ，便不做上述处理，而按表 B 所提供的联址，查访另一个下结点；当联组 I 的所有下结点都经过查访并予以处理后，就将 $A(I, 1)$ 的标志置 0，表示查访完毕。这个过程反复进行，最后在表 A 的第三格中形成最长的行程链（即最大风压值线路）。

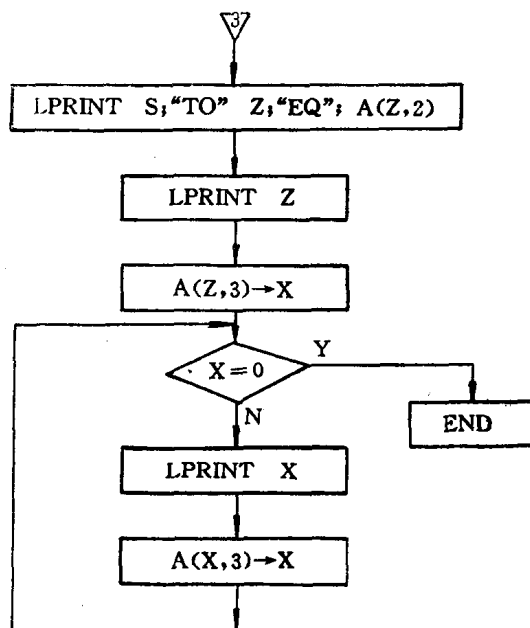


图 1-6

(5) 打印结果。结果打印由图1-6所示程序段执行。

以上面介绍的例子为例，根据这个程序段打印出来的结果与格式是这样的：

1 TO 6 EQ 33 从起点 1 到终点 6 的风压值为 33；

(6)—(5)—(3)—(1) 线路为 1—3—5—6 这条线。

上面讨论的是各段的具体分析处理，这些工作做完后，则可把各段用 BASIC 语句组合起来，构成程序。

4. 按照 BASIC 的语法规则，把这个过程用不同语句进行组合，构成程序

这一工作比较简单，就是用语句将上面考虑的整个过程表达出来。需要注意的地方是各段的衔接处，这在具体编程中应具体处理。此例的程序见后。

二、程序的使用说明

1. 基础资料

在使用该程序以前,应具备以下几种基础资料:

- (1) 根据通风系统图,构造出通风网络图;
- (2) 在网络图上,应按顺序填写各支路的结点编号,并填入各支路对应的风压。

2. 数据准备

上机前,应根据网络图提供的数据,确定出 N 、 M 、 S 、 Z 各值,并把各支路的起点 A 、终点 B 及其对应的风压值一一罗列于纸上,以供上机使用。

3. 上机计算

- (1) 输入源程序(若已寄存,则调用);
- (2) 把 N 、 M 、 S 、 Z 按65语句所要求的格式输入;
- (3) 按170语句要求,输入 A 、 B 、 R 一组数(每一循环仅输入一组数,直至循环结束);
- (4) 打印出的结果则为所求的最大通风线路及其阻力值。

该程序的使用是很简单的,只要输入程序及按要求输入各数据,则可很快地求出所要求的通风线路及风压值。

三、源程序

```

60 DIM A(40,4),B(80,3)
65 INPUT N,M,S,Z
66 PRINT"N=";N;TAB 10;"M=";
    M;TAB 20;"S=";S;TAB 30;
    "Z=";Z
70 FOR I=1 TO N
80 A(I,1)=1
90 A(I,2)=0
100 A(I,3)=0
110 A(I,4)=0
120 NEXT I
130 A(S,2)=0
160 FOR P=1 TO M
170 INPUT A,B,R
175 PRINT A,B;R
180 B(P,2)=B
190 B(P,3)=R
200 B(P,1)=A(A,4)
210 A(A,4)=P
220 NEXT P
230 I=S
240 GOTO 320
250 I=I+1
260 IF I<=N THEN 310
270 I=1
280 K=K+1
290 IF K>I THEN 450
310 IF A(I,1)=0 THEN 250
320 J=A(I,4)
330 K=0
340 T=A(I,2)+B(J,3)
350 L=B(J,2)
360 IF A(L,2)>=T THEN 400
370 A(L,1)=1
380 A(L,2)=T
390 A(L,3)=I
400 J=B(J,1)
410 IF J<>0 THEN 340
420 A(I,1)=0
430 GOTO 250
450 LPRINT S;"TO";Z;"EQ";A(Z,
    2)
460 LPRINT("(";Z;")--";
470 X=A(Z,3)

```