广西科学院学报
Journal of Guangxi Academy of Sciences

2010,26(4):429~432
Vol. 26, No. 4 November 2010

闰年判断的算法表达式及其应用实例
Leap Year Judgement Expression and Its Application Examples

李世才[1],苗 丽[2],劳 薇[1]
LI Shi-cai[1], MIAO Li[2], LAO Wei[1]

(1. 广西南宁水利电力设计院,广西南宁 530001;2. 广西经贸职业技术学院,广西南宁 530021)
(1. Guangxi Nanning Water Conservancy and Electric Power Designing Institute, Nanning, Guangxi, 530001, China; 2. Guangxi Economic & Trade Polytechnic, Nanning, Guangxi, 530021, China)

摘要:运用天文、历法原理和数学连分数理论,通过一些数学变换得到求解闰年问题的一种不含条件判断的算法表达式,并应用算法表达式计算水能、任意2个日期的相隔天数和任意一天是星期几.结果显示,该算法表达式计算步骤简便,求解灵活快速,通用性较强.

关键词:算法 闰年 平年 回归年 公历
中图法分类号:TP312,O156.1 **文献标识码**:A **文章编号**:1002-7378(2010)04-0429-04

Abstract: Based on astronomy, calendar principles and mathematical theory of continued fraction, by several mathematical transformations, a method of algorithm expression without conditional judgement solved the leap year problems. Three application examples in calculating hydroenergy, the days between any two dates and the week for some days were presented. The advantages of algorithm expression includes simple calculation steps of algorithm, flexible and swift solving, great commonality, etc.

Key words: algorithm, leap year, common year, tropical year, Gregorian calendar

在水文测验资料整编、水文分析与计算、水利计算、日调节与径流式水电站的水能计算、气象资料统计分析等工作,以及我们的日常生活里,特别是一些与日期和时间有着密切关系的重要部门,经常需要准确判断某年是否为闰年或者某个年段内的闰年个数,以期确定其2月份的天数.对于公历中每逢4年一闰是指能被4整除的年是闰年(不能被4整除的年就是平年),但是逢百之年,能被4整除的并不都是闰年,必须还要能被400整除的才是闰年[1~5].

显然我们可以根据历法原理用条件表达式来判断某年是否为闰年[1~6],但是当所判断年份的范围较大(比如制作万年历)时[7],由于还要考虑其能否被3324(能被3324整除的年也能被4整除,但它不是闰年)整除的问题,所以用条件表达式判断闰年,其逻辑结构就比较复杂,编制电算程序时容易漏掉某些条件而使结果出错,并且不易看出错误.现已报道的有关闰年算法的表达式,都存在着通用性不够强、使用条件限制多等问题.根据天文学、历法原理、算法语言和数学的相关知识[1~5],利用一些数学变换可以得到求解上述问题而不含条件判断的一种算法表达式.经过近25年的电算实践证明,该算法表达式具有计算步骤简便,求解灵活快速,通用性强等优点,是一种比较理想和实用的算法表达式.

收稿日期:2010-08-28
修回日期:2010-09-02

作者简介:李世才(1956-),男,高级工程师,主要从事工程水文分析、水能规划设计及相关算法的研究;苗 丽(1982-),女,助教,主要从事计算机网络、图形图像的研究与教育;劳 薇(1981-),女,工程师,主要从事计算机技术的应用与管理。

1 闰年判断原理及算法表达式

天文学上把地球绕太阳一周从春分点回到春分点的时间称为一个回归年,其长度约为365.2422个太阳日[1],相当于一年有365天,为了弥补上述小数

部分带来的误差，历法就规定利用闰年来补足其差数. 地球绕太阳公转一周需365天5小时48分46秒，也就是[2~4]

$$365+\frac{5}{24}+\frac{48}{24\times 60}+\frac{46}{24\times 60\times 60}=365\frac{10463}{43200}\text{(天)}. \tag{1}$$

根据辗转相除法[2~4]，式(1)展开为连分数得

$$365\frac{10463}{43200}=365+\frac{1}{4+}\frac{1}{7+}\frac{1}{1+}\frac{1}{3+}\frac{1}{5+}\frac{1}{64}. \tag{2}$$

式(2)分数部分的渐近分数是

$$\frac{1}{4},\frac{7}{29},\frac{8}{33},\frac{31}{128},\frac{163}{673},\frac{10463}{43200}.$$

这说明4年加1天是初步的最好逼近. 但是29年加7天更精密些，33年加8天又精密些(也就是99年加24天，我们的算法是100年加24天)；128年加31天更精密(这就是说，头3个33年加8天，后一个29年加7天，共29+33×3=128年加31天. 在400年内，有3个128年，4个4年，所以400年加31×3+1×4=97天，这与我们的算法相同). 这就是历法订定每4年一闰，而每100年少一闰，每400年又加一闰的算法来源[2~4].

如果按照上述的算法计算，当经过43200年后，就应该有43200÷4−43200÷100+43200÷400=10476个闰年，但是按照精密的计算，结果却只有10463个闰年，即应该加10463天，这样总共多加了13天，照此制作万年历就会出现大的差错[7]. 为了弥补其差数，则每隔43200÷13=3323.077年就应该减1天，根据所要达到的目的，将式(1)右边的分数部分拆为4个单位分数[8]之和(差)为

$$\frac{10463}{43200}=\frac{1}{4}-\frac{1}{100}+\frac{1}{400}-\frac{1}{X}. \tag{3}$$

解式(3)中的分式方程可得 $X\approx 3323.077$. 由于年份必须为正整数，还要求 X 能被4整除并与3323.077最为接近，故式(3)中的 X 应该取值为3324[7]. 这样用式(3)的右边代替式(3)的左边其误差小于 8.3567322×10^{-8} 天(1年的误差)，因此当年数大于3323年时，历法还须订定每3324年又减一闰，就是到了43200年其误差也不超过 3.6101×10^{-3} 天，到了345700年其误差也不超过0.03天，这样我们的历法才精确. 故有

$$\frac{10463}{43200}\approx\frac{1}{4}-\frac{1}{100}+\frac{1}{400}-\frac{1}{3324}=\frac{80507}{332400}. \tag{4}$$

我们可以将已知的年份分别除以式(4)中间4个分数的分母，而根据其整除关系来判断该年是否为闰年，并利用条件表达式来设计程序. 但是由于所用的条件表达其逻辑结构比较复杂[7]，故编制电算程序也是一件不容易的事情. 我们探讨不用条件表达式判断而用一个算法表达式比较简便地表示出任意一年的2月份天数. 以 N 代表任意一年年号，用符号函数 SGN(X)和取整函数 INT(X)表示2月份天数的表达式为

$$T=28+\sum_{i=1}^{4}(-1)^{i+1}\mathrm{SGN}\left(\mathrm{INT}\left(\frac{N}{A_i}\right)-\frac{N}{A_i}\right). \tag{5}$$

式(5)中的 $A_1=4, A_2=100, A_3=400, A_4=3324$，该式保留或舍去的分别对应于 N 能被4、100、400、3324整除的一天.

2 算法表达式的应用实例

2.1 水能计算

用GWBASIC语言和式(5)编制程序段，用来处理数据的输入和存储问题，其部分程序如下(为节省版面，一行中留尽可能多的语句，并且符合其语法要求)：

```
5 OPTION BASE 1:DEFINT A-Y:READ N1,N2,Y1,Y2,YU
10 M=N2-N1:Y=YU:IF YU=1 THEN M=M+1
15 DIM A(4),B(12),T(M):A(1)=4:A(2)=100:A(3)=400:A(4)=3324
20 FOR I=1 TO 12:IF Y>12 THEN Y=1
25 IF Y=4 OR Y=6 OR Y=9 OR Y=11 THEN B(I)=30:GOTO 35
30 IF Y=2 THEN YY=I ELES B(I)=31
35 IF Y=Y1 THEN J1=I
40 IF Y=Y2 THEN J2=I
45 Y=Y+1:NEXT I:Y=N1:IF YU<3 THEN Y=Y-1
50 FOR I=1 TO M:C=-1:Y=Y+1:T=28
55 FOR J=1 TO 4:C=-C:Z=Y/A(J):T=T+C*SGN(INT(Z+1E-12)-Z):NEXT J
60 T(I)=T:NEXT I:FOR I=1 TO M:PRINT T(I);",";:NEXT I:PRINT
65 INPUT "A$=";A$:A$="E:"+A$:OPEN "O",#1,A$:N=0
70 FOR I=1 TO M:B(YY)=T(I):FOR J=1 TO 12
```

```
75 FOR K=1 TO B(J):READ Z:PRINT #1,Z;:
N=N+1:NEXT K
80 NEXT J:NEXT I:CLOSE:PRINT "N=";N
85 FOR I=0 TO N:PRINT I,Z:READ Z:NEXT
I:END
```

程序段中:N1 为所输入资料的起始年号,N2 为终止年号;Y1 为一年中丰水期的开始月份,Y2 为终止月份;YU 为输入第一年的第一个月资料时的开始月份,当 YU=1 时,表示按日历年统计计算,当 YU>1 时,表示按水文(或水利、灌溉等)年统计计算.行号 5 是先指定各数组变量的下标从 0 开始,并定义字母 A～Y 开头的所有变量都是整型量,然后读出变量 N1,N2,Y1,Y2,YU 的值;20～45 行号用来给每个月份的天数赋值,2 月份的天数还不确定时,30 行号先记住其在一年中的位置序号;45～60 行号用来计算(根据式(5))从第 N1 年到第 N2 年这个年段内的所有年号的 2 月份天数,并把结果显示或打印出来;65 行号是先输入数据文件的名称,然后在磁盘 E 上建立磁盘顺序文件;70～80 行号用来实现将某水文站的资料(这里是指逐日平均流量)按逐年(月、日)的顺序读出数据,并逐一用写语句将这些数据存储在计算机的磁盘中,也可将 65～80 行号的语句稍作修改,把这些数据按丰水期和枯水期两个系列的文件存储在磁盘中;85 行号是用来检查有无多输入的数据;在实际工程的计算时,又需要将存储在磁盘中的数据按顺序逐一读出,这只需将 65 行号中的 OPEN 语句后的"O"改为"I",75 行号中的写语句 PRINT 改为读语句 INPUT 就行.

2.2 任意 2 个日期相隔天数的计算

解决此问题的关键是正确计算出任意一个日期,从公元的纪年(1 年 1 月 1 日)开始到已知日期的实际总天数 TZ,已知日期的年号是 N,月份是 M,日数是 R,根据式(1)和式(4)可得[1]

$$TZ=365(N-1)+\left[\frac{N-1}{4}\right]-\left[\frac{N-1}{100}\right]+\left[\frac{N-1}{400}\right]-\left[\frac{N-1}{3324}\right]+C. \tag{6}$$

由于 TZ 为正整数,所以式(6)中的 4 个分式的值也必须取其整数部分,用符号[ ]表示.编制 BASIC 语言程序时,用取整函数 INT(X)来实现,其中的 C 是从已知这一年 N 的元旦计算到已知日期日数 R 这天为止(包括这一天也在内)的天数,并且有 $C=\sum_{i=0}^{M-1}B_i+R$,其中 $B_0=0$, B_1, B_2, …, B_{11} 分别是 1 月,2 月,…,11 月份的天数.从 C 的表达式可看出,最关键的是计算 2 月份的天数,2 月份的天数可用式(5)计算.为方便起见,这里先假定 $B_2=28$ 天,由于每个月份的天数都不会少于 28 天,所以可先把每个月份的天数都当成 28 天,然后只累计其差数就行了,其结果见表 1.

表 1 差数 B_i-28 按序累计结果 C_i 排列表

月份	B_i	B_i-28	差数累计 C_i	月份	B_i	B_i-28	差数累计 C_i
0	0	0	0	6月	30	2	13
1月	31	3	3	7月	31	3	16
2月	28	0	3	8月	31	3	19
3月	31	3	6	9月	30	2	21
4月	30	2	8	10月	31	3	24
5月	31	3	11	11月	30	2	26

从表 1 可以看出,当 N 为平年时, C_i 为上述表 1 中的值;当 N 为闰年时, C_i 应从 C_2 开始由上述表 1 中的值加上 1.也就是说平年时加上 0,闰年时应加上 1,这可用式(5)中右边和号部分来实现,它的值不是 0 就为 1,用符号 T_{01} 表示.我们还发现当 $M>2$ 时,C_i 才加上 T_{01},否则 C_i 保持不变,可以用乘 1 或乘 0 来实现.在 GBASIC 语言系统中关系表达式($M>2$)成立时,其值为 -1,否则(不成立时)其值为 0,对关系表达式($M>2$)取绝对值后其值就变成了 1 或 0,即 ABS($M>2$)=1 或 0.根据上述的分析过程,可得到式(6)的一种通用算法表达式:

$$TZ=365(N-1)+28(M-1)+\sum_{i=1}^{4}(-1)^{i+1}\mathrm{INT}\left(\frac{N-1}{A_i}\right)+C_{M-1}+\mathrm{ABS}(M>2)\times T_{01}+R. \tag{7}$$

A_i 的取值与式(5)相同,式(7)中的 T_{01} 为

$$T_{01}=\sum_{i=1}^{4}(-1)^{i+1}\mathrm{SGN}\left(\mathrm{INT}\left(\frac{N}{A_i}\right)-\frac{N}{A_i}\right). \tag{8}$$

根据上述的数学模型可比较方便地设计出其算法程序(这里省略).

2.3 任意一天是星期几的计算

根据式(7)中右边的第一项有 $365(N-1)=7\times52(N-1)+N-1$,此式右边的第一项是 7 的倍数,能被 7 整除,故这一项可以省去.又因为 28 是 7 的倍数,能被 7 整除,故又可以将式(7)中右边的第二项省去;而且 ABS($M>2$)$\times T_{01}$ 的值不是 1 就为 0,它的值由 N、M 的值决定,主要根据式(8)中的表达式计算.所以按照下面的公式计算,就可以求出某年 N 月 M 日 R 是星期几.

$$W=N-1+\sum_{i=1}^{4}(-1)^{i+1}\mathrm{INT}\left(\frac{N-1}{A_i}\right)+C_{M-1}+\mathrm{ABS}(M>2)\times\sum_{i=1}^{4}(-1)^{i+1}\mathrm{SGN}\left(\mathrm{INT}\left(\frac{N}{A_i}\right)-\frac{N}{A_i}\right)+R. \quad (9)$$

式(9)中 A_i 的取值与式(5)相同，其中 C_{M-1}($M=1,2,3,\cdots,12$)的取值见表1中的 C_i 值或下面程序段数组变量 C(I)的值，SGN(X)是符号函数，INT(X)是取整函数，ABS(X)是绝对值函数. 求出 W 以后，用7除，通过余数就知道这天是星期几. 用 GWBASIC 语言和式(9)编制的程序段如下(有关变量的说明部分已省去)：

```
100 DIM A(4),C(11):A(1)=4:A(2)=100:A(3)=400:A(4)=3324:CLS
105 C(0)=0:C(1)=3:C(2)=3:C(3)=6:C(4)=8:C(5)=11
110 C(6)=13:C(7)=16:C(8)=19:C(9)=21:C(10)=24:C(11)=26
115 INPUT "N=";N:INPUT "M=";M:INPUT "R=";R:W=N-1:T=0:C=-1
120 FOR I=1 TO 4:C=-C:Z=N/A(I):W=W+C*INT((N-1)/A(I)+1E-12)
125 T=T+C*SGN(INT(Z+1E-12)-Z):NEXT I
130 W=W+C(M-1)+R+ABS(M>2)*T:XQS=W-7*INT(W/7+1E-12)
135 PRINT "T=";T:PRINT "XQS=";XQS:GOTO 115
```

3 算法比较

就求解任意一天是星期几的问题与蔡勒(Zeller)公式做比较：

$$W=Y+\left[\frac{Y}{4}\right]+\left[\frac{C}{4}\right]-2C+\left[\frac{26(M+1)}{10}\right]+D-1. \quad (10)$$

式(10)中，W 是星期数，Y 是年份后两位数字，C 是世纪数减1，M 是月份，D 是日数，其中要把某年的1月、2月当成上一年的13月、14月来算，这时 Y 和 C 均按上一年取值，符号[ ]表示取整数部分，W 的最后取值方法与2.3节相同.

从式(9)和式(10)的表达式看，式(10)由于有了一些特别的规定，比如1月、2月要当成上一年的13月、14月来算(主要是为避免计算2月份的天数)，所以使用起来就显得有一些别扭，并且还要作一些特殊的处理，所以编制算法程序比式(9)显得麻烦. 并且式(10)只适用于1582年10月15日之后的计算，当年号 $N>3323$ 年时，其计算结果不正确. 例如3324年10月1日，式(9)计算的结果是平年和星期六，式(10)计算的结果是闰年和星期日，后者比前者多算了1天. 对于9999年1月1日，式(9)计算的结果是平年和星期二，式(10)计算的结果是平年和星期五，后者比前者多算了3天. 这是由于式(10)的推导过程中没有考虑每逢3324年又减一闰的规定所导致的. 式(9)适用于公元1年1月1日之后到公元1千万年左右，其最大误差也不超过1天(到那时地球绕太阳公转一周的时间有可能已经发生了较大的变化).

4 结束语

20世纪90年代末期，早期的计算机系统在处理日期的表达和闰年的计算时，为了节省有限的内存空间，只进行了简单的处理(例如年号只处理后两位数字，没有完全考虑逢100年减1闰和逢400年加1闰的问题). 又由于当时已基本上进入初级的信息化时代，并接近世纪之交，早期存储的海量信息需要进行交换和处理，而新老系统对接的软件还没有研制出来，所以引起了全世界的恐慌(即所谓的“千年虫”问题). 现有一般的计算机系统在处理日期的表达和闰年的计算时，也还没有考虑到每逢3324年又减1闰的这一规定，为防患今后出现“3324年虫”和“万年虫”的问题，和为需要这方面做精密计算的部门提供方便来考虑，本文的研究和所提供的算法表达式都具有比较重要的理论意义和使用价值，完全能够满足工程设计和一般科学研究的计算需要，值得推广采用.

参考文献：

[1] 十万个为什么(1)[M]. 第二版. 上海：上海人民出版社，1972：92-97.

[2] 华罗庚. 从祖冲之的园周率谈起(数学小丛书3)[M]. 北京：科学出版社，2002：1-23.

[3] 华罗庚. 从孙子的“神奇妙算”谈起——数学大师华罗庚献给中学生的礼物(华罗庚专辑)[M]. 北京：中国少年儿童出版社，2006：68-87，129-133.

[4] 华罗庚. 高等数学引论(第一册)[M]. 北京：高等教育出版社，2009：34-35.

[5] 汪燮华，郁宝忠，余晓清，等. 计算机基础[M]. 上海：上海科学技术文献出版社，1984：106-110.

[6] 谭浩强. BASIC 趣味程序选(三)[M]. 北京：清华大学出版社，1985：106-109.

[7] 谭浩强. BASIC 趣味程序选(四)[M]. 北京：清华大学出版社，1987：289-291.

[8] 柯召，孙琦. 单位分数(数学小丛书14)[M]. 北京：科学出版社，2002：1-64.

(责任编辑：尹 闯)