## 简化幂数法处理加法链问题

## 摘要：本文针对一个正整数的加法链问题，基于幂树法及二进制方法，综合分析，给出了一种较为简便、高效的算法，能够快速求出加法链，虽不能保证求得的加法链是最短加法链，但能取得长度较短的加法链，可用于快速确定加法链上限。

## 关键词：加法链；幂树法；二进制方法；简化幂树法

## ABSTRACT: Based on the power tree method and binary method, this paper gives a simple and efficient algorithm for the problem of adding chain of a positive integer. It can find the adding chain quickly. Although it can't be guaranteed that the adding chain is the shortest, it can get the adding chain with shorter length, which can be used to determine the upper limit of the adding chain.

## Key words: addition chain; power tree method; binary method; simplified power tree method

## 1 引言

给定一个正整数*n*，一个长度为*r*的可计算*n*的加法链(addition chain) *U* 是一个严格递增的正整数序列，其中，且对任意的，是它前面两个元素(不必不同)的和，即存在 使得，当时，称该步为倍乘。当然，找到的可计算*n*的加法链长度越短越好，但对于给定的*n*，找一个具有最短加法链长度的加法链是很困难的问题。

## 2 预备知识

幂树法是解决加法链问题的基础方法，通过穷举搜索确定出最短加法链。求最短加法链时，用幂树法求解较小的数结果较好，对于较大的数，幂树法的时间空间开销太大，原因在于，当深度为*k*时，幂树加深一层，考虑该层子结点，将有*k*个子结点；考虑全部结点，这样将有个结点，开销很大。可用的解决办法是用剪枝函数，把某些不合条件的枝减去。但在能保证得到的是最短加法链的情况下，构造剪枝函数，其效率虽然可以提高不少，但对于这样的量级，当*k*较大时还是难以计算。

二进制方法是一种处理加法链问题的简便的方法，通过将一个正整数表示成二进制形式来分析，如果对应位上为1，就把对应的数加入加法链中，所有位相加就得到了目标数。二进制方法比较简便，但求出的不是最短加法链，更常用于作为最短加法链的上界。

基于上述两种方法，本文给出一种求解加法链问题的方法。

## 3 简化幂树法

**3.1 最简单的简化幂树**

考虑得到一个数的加法链，当选用越多的倍乘时，其加法链越有可能短，于是用一种极限的想法：

考虑构建一棵树，根结点是1，之后每个主干结点都是2的若干次方，在主干结点下，仅有其加1后乘以2的若干次方的链，如图 1所示。

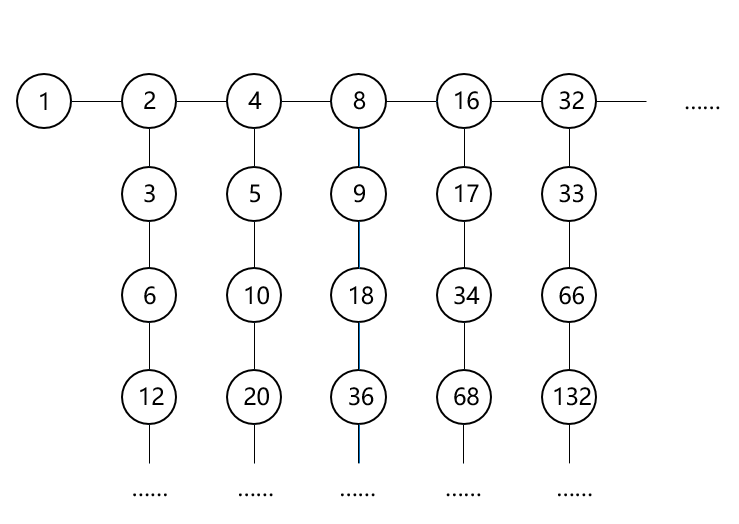


图 1 示例树

求的加法链算法描述：

(1) 先走主干结点，能直接得到则记录当前链，否则当走到大于的时候，就往回逐个加，可以得到，记录当前链；

(2) 从第一条链开始，同样的，往下走，能直接得到则记录当前链，否则往回逐个加，可以得到，记录当前链；

(3) 走完所有链，将记录的加法链长度的最小值对应的链输出。

算法的具体过程如下：

|  |
| --- |
| 最简单的简化幂树法 |
| addChain(n): k=ln(n)/ln(2); search(1,2,4…); for(i from 1 to k):  search(1,2,4…,2i,2i+1,(2i+1)\*2, (2i+1)\*4…);  search(a[]): if (r:a[r]==n) return(1,2,4…,2r); else r: a[r]<n and a[r+1]>n  for(i from r-1 to 0):  if（a[r]+a[i]<n）add a[i];   else if（a[r]+a[i]==n）return a; |

**3.2 一般的简化幂树**

最简单的简化幂树可以看作对二进制方法的改进，使得每次1的增加数目为2，这部分可以推广至一般，使每次1的增加数目为3，4等，比如使每条支链的第一个节点为主链节点+3，+7等等；同时增加时，1的位置同样需要考虑，如同样每次1的增加数目为3，可以是+3，也可以是+5，+9等。

## 4 算法分析

简化幂树法是高效的，可以计算出主链节点数是，每条支链的节点数为，因此时间复杂度是，空间上由于可以对前链覆盖，所以空间复杂度为*O()*。且由于主链搜索即满足二进制方法，因此每个数必然都至少能通过搜索主链来生成。

算法缺点如二进制方法一样，无法保证求得的加法链是最短加法链。但不同于二进制方法和最短加法链差距会很大，简化幂树法能保证结果更接近于最短加法链。

该算法最坏的情况是退化为二进制方法，对正整数，记记，表示的二进制表示中1的个数，为的最短加法链长度。

最坏的情况下，此算法退化为二进制方法。根据二进制方法，加法链长度为。

最好的情况下，是可分解为，使得每次1的增加数目为，此算法加法链长度为，当且仅当时等号成立。

故此算法取得的加法链范围为：

。

一般情况下，此算法将找到一个，可分解为，使得每次1的增加数目为，加法链长度为。

下面是对于10000以下的正整数用简化幂树法求解和对应的最短的加法链[1]的差距：

表 1 最简单的简化幂树结果与实际最短比较

|  |  |  |
| --- | --- | --- |
| 差距 | 数目 | 比重 |
| 0 | 4247 | 0.4247 |
| 1 | 4743 | 0.4743 |
| 2 | 958 | 0.0958 |
| 3 | 51 | 0.0051 |
| 4 | 1 | 0.0001 |

表 2 一般简化幂树结果与实际最短比较

|  |  |  |
| --- | --- | --- |
| 差距 | 数目 | 比重 |
| 0 | 5735 | 0.573500 |
| 1 | 4154 | 0.415400 |
| 2 | 110 | 0.011000 |
| 3 | 1 | 0.000100 |

可以看出，在此范围内，使用一般的简化幂树时，求得的结果中最短或次短（比最短的结果多1）的比重达到98.89%，表明能取得较好的结果。

## 5 总结

简化幂树法改进自二进制方法，形式上是极为简化的幂树，已不能保证得到的加法链是最短的结果，但将结点总数由缩减到，将大大加快速度，同时能够取得更接近甚至是最短的结果，适合于快速确认上限，这对于一些深度加深时难度急剧增加的算法，具有重要作用。

## 【参考文献】

[1] http://wwwhomes.uni-bielefeld.de/achim/addition\_chain.html