International Journal of Scientific & Engineering Research, Volume 6, Issue 5, May-2015
ISSN 2229-5518

Some algorithmic methods for computing the sum of powers.

Yerzhan Utkelbayev, Madiyar Aitbayev

May, 2015

Abstract

In this paper several methods with different algorithmic complexity are considered for sum of powers. Different algorithmic methods are shown based on some known mathematical facts.

1 Introduction

Suppose we have positive integers numbers n, k and p. Find:
$f(n,k) = 1^k + 2^k + ... + n^k = \sum_{i=1}^{n} i^k$.

Sum of powers were investigated in 17th Century by Johann Faulhaber of Ulm. He described sum of powers in terms of $n(n+1)/2$. D. Knuth showed [1] that Faulhaber got this result for sum of the $13th$ powers:

$$\frac{960N^7 - 2800N^6 + 4592N^5 - 4720N^4 + 2764N^3 - 691N^2}{105}, where$$

N= $\frac{n(n+1)}{2}$

He also found closed formulas for some small $13 <= k <= 17$ and states that there should be polynomials with alternating signs for all sum of powers [1].

Nowadays, it calls Faulhaber's formula. It can be expressed as sum of powers $(k+1)$th-degree polynomial function of n with Bernoulli numbers [2]:
$\sum_{k=1}^{n} k^p = \frac{1}{p+1} \sum_{j=0}^{p} (-1)^j \binom{p+1}{j} B_j n^{p+1-j}$, where $B_1 = -\frac{1}{2}$.

Interesting facts which can help calculate sum of powers by modulo p (prime number) were provided by Kieren MacMillan, Jonathan Sondow[3].

For the first kth formulas:

$k = 1$, $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} = \frac{n^2+n}{2}$

k = 2, $1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} = \frac{2n^3+3n^2+n}{6}$

k = 3, $1^3 + 2^3 + 3^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^4+2n^3+n^2}{4}$

k = 4, $1^4 + 2^4 + 3^4 + \cdots + n^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$

International Journal of Scientific & Engineering Research, Volume 6, Issue 5, May-2015
ISSN 2229-5518

$$= 6\text{n}\frac{^5+15n^4+10n^3-n}{30}$$
$$k=5,\ 1^5+2^5+3^5+\cdots+n^5=\frac{n^2(n+1)^2(2n^2+2n-1)}{12}$$
$$= 2\text{n}\frac{^6+6n^5+5n^4-n^2}{12}$$

$$k=6, 1^6+2^6+3^6+\cdots+n^6=\frac{n(n+1)(2n+1)(3n^4+6n^3-3n+1)}{42} \quad (1)$$
$$=\frac{6n^7+21n^6+21n^5-7n^3+n}{42} \quad (2)$$

2 Methods

Method 1.

Using binomial coefficient formula it is know that $(n+1)^k=\sum\limits_{i=0}^{k}\binom{k}{i}n^i$ (1).

Let's call $s_i = 1^k+2^k+...+i^k=\sum\limits_{j=1}^{i}j^k$.

Next relation can obtained by using formula (1):

$$\begin{pmatrix}\binom{k}{0} & \binom{k}{1} & \cdots & \binom{k}{k} & 0\\ 0 & \binom{k-1}{0} & \cdots & \binom{k-1}{k-1} & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \binom{k}{0} & \binom{k}{1} & \cdots & \binom{k}{k} & 1\end{pmatrix}\begin{pmatrix}i^k\\ i^{k-1}\\ \vdots\\ s_i\end{pmatrix}=\begin{pmatrix}\binom{k}{0}*i^k+\binom{k}{1}*i^{k-1}+\cdots+\binom{k}{k}*i^0+0*s_i\\ \binom{k-1}{0}*i^{k-1}+\cdots+\binom{k-1}{k-1}*i^0+0*s_i\\ \vdots\\ \binom{k}{0}*i^k+\binom{k}{1}*i^{k-1}+\cdots+\binom{k}{k}*i^0+1*s_i\end{pmatrix}$$
$$=\begin{pmatrix}(i+1)^k\\ (i+1)^{k-1}\\ \vdots\\ s_{i+1}\end{pmatrix}$$

Let's call

$$\text{A} = \begin{pmatrix}\binom{k}{0} & \binom{k}{1} & \cdots & \binom{k}{k} & 0\\ 0 & \binom{k-1}{0} & \cdots & \binom{k-1}{k-1} & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \binom{k}{0} & \binom{k}{1} & \cdots & \binom{k}{k} & 1\end{pmatrix}$$

By using above relation we can make next calculations:

$$\text{A}^n\begin{pmatrix}1^k\\ 1^{k-1}\\ \vdots\\ s_1\end{pmatrix}=\begin{pmatrix}n^k\\ n^{k-1}\\ \vdots\\ s_n\end{pmatrix}$$

Matrix multiplication of two matrices size of $k*k$ can be done in O(k^3). Matrix multiplication is associative. Therefore using fast multiplication and above formula sum of powers can be computed in complexity O($k^3 log(n)$).

2

International Journal of Scientific & Engineering Research, Volume 6, Issue 5, May-2015
ISSN 2229-5518

Method 2.

We can use divide and conquer algorithm [4][5] recursively:
if n is odd then $f(n,k) = f(n-1,k) + n^k$

if n is even then $f(n,k) = f(n/2,k) + (n/2+1)^k + (n/2+2)^k + ... + (n/2+n/2)^k = f(n/2,k) + \sum_{i=1}^{n/2}(n/2+i)^k = f(n/2,k) + \sum_{i=0}^{k}(\binom{k}{i}f(n/2,i)(n/2)^{k-i})^k$.
if $n = 1$ then $f(n,k) = 1$

We can precalculate binomial coefficients in O(k^2) using it's recursion formula $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. The recursion with different parameters should be called $klogn$ times. We will use memorization method for not solving one recursion two times. One recursion call works in O(k). So the overall complexity of this algorithm is O($k^2 log(n)$).

Method 3.

As previously mentioned in general case $f(n,k)$ is polynomial with degree $(k+1)$. Let's find coefficients of polynomial efficiently. Using Lagrange's interpolation it can be done in complexity $O(k^2)$. In this problem values different at $k+2$ points needed. We can calculate at first $k+2$ points, in other way, f(i, k) for $1 <= i <= k+2$. So, for this part the complexity will be $O(k)$.

Finally, using Lagrange's polynomial interpolation values at $k+2$ different points we can recover coefficients of f(n, k). Total complexity: $O(k^2)$.

3 Conclusion

In the table below we can compare methods listed before:

Method	Description	Complexity
1	matrix multiplication	O($k^3 log(n)$)
2	Divide and conquer	O($k^2 log(n)$)
3	Lagrange's polynomial interpolation	O(k^2)

It is worth mentioning that we take complexity as O(1) of the operation with two numbers such as multiplication, addition, subtraction and division. The most efficient method in the list is Lagrange's polynomial interpolation.

4 References

1. Donald E. Knuth (1993). "Johann Faulhaber and sums of powers". Math. Comp. (American Mathematical Society) 61 (203): 277-294.

2. John H. Conway, Richard Guy (1996). The Book of Numbers. Springer. p. 107.

3. Kieren MacMillan, Jonathan Sondow (2011). "Proofs of power sum and binomial coefficient congruences via Pascal's identity". American Mathematical Monthly 118: 549-551.

4. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, Introduction to Algorithms (MIT Press, 2000)

5. Brassard, G. and Bratley, P. Fundamental of Algorithmics, Prentice-Hall, 1996.