Descrição de Algoritmo sublinear em Tempo para Cálculo de Fatoriais no Contexto da Aritmética Modular

Rafael Granza de Mello – rafaelgranzademello@gmail.com

UDESC - Centro de Ciências Tecnológicas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação - Integral

30 de Outubro de 2022

1 Introdução

Durante os meus estudos relacionados à maratona de programação, compreendi que os competidores precisam conhecer muitos algoritmos avançados que, muitas das vezes, são o estado da arte para a resolução de determinados problemas. No decorrer dos meus anos como competidor, aprendi muito sobre o funcionamento de diversos algoritmos e análise de complexidade.

Um dos problemas mais marcantes que encontrei era o simples cálculo de um:

$$n! \mod p \tag{1}$$

Embora pareça simples à primeira vista, como p é primo e os limites são de $1 \leq n, p \leq 10^{11}$ [2], é evidente que uma implementação mais elaborada para a obtenção do resultado em poucos segundos seria necessária. A solução desse problema requereu o uso de diversos tópicos interessantíssimos como *Fast Fourier Transform* (FFT) e *Number-theoretic Transform* (NTT), aritmética modular aplicada em polinômios, teorema do resto chinês sobre polinômios e divisão e conquista.

Apesar de ser um problema conhecido entre os competidores de alto-desempenho da maratona de programação, o material para a descrição deste algoritmo está disperso entre vários artigos, ou então espalhado em blogs e fóruns pela internet [5][1]. Dito isso, é imprescindível uma descrição mais formal desse algoritmo.

2 Contextualização

Para a construção deste trabalho, é necessário apresentar de forma sucinta alguns dos conceitos que serão utilizados para fundamentar a pesquisa.

No contexto proposto do problema descrito anteriormente, pode-se assumir que os resultados cabem em variáveis de tamanho fixo, dessa forma as operações aritméticas simples tem custo constante, *i.e.* $O(1)$.

2.1 Fatorial

Apesar de ser uma das funções mais famosas da matemática e dispensar apresentações, é primordial defini-lo. A descrição mais natural é dada por [4]

$$n! = \prod_{i=1}^{n} i \tag{2}$$

Para os fins deste trabalho, serão feitas representações do fatorial em forma de polinômios. Uma das formas mais naturais de estender essa representação é por meio de um polinômio P_n de grau $n-1$ definido como:

$$P_n(x) = \prod_{i=1}^{n} (x+i) \tag{3}$$

Para o qual temos que:

$$P_n(0) = n! \tag{4}$$

Utilizando o método de Horner, a Equação 4 pode ser computada em tempo $O(n)$. Contudo, ainda não é suficiente para a implementação de um algoritmo que tenha tempo sublinear.

Para isso, será necessário se valer da técnica de decomposição em raiz quadrada. Desse mesmo modo, definiremos outro polinômio Q_n de grau $\sqrt{n}$ definido como:

$$Q_n(x) = \prod_{i=1}^{\sqrt{n}} (x+i) \tag{5}$$

Para valores de n que sejam quadrados perfeitos temos que:

$$\prod_{i=0}^{\sqrt{n}} Q_n(i) = n! \tag{6}$$

Para ampliar esse resultado para valores de n que não sejam quadrados perfeitos, basta calcular o fatorial do maior quadrado perfeito menor que n. Denominando esse valor por $n' = \lfloor \sqrt{n} \rfloor^2$, tem-se que o polinômio $Q_{n'}$ pode ser usado para o computar $n'!$, e que esse resultado multiplicado pelos números inteiros no intervalo $[n'+1, n]$ é justamente $n!$. No Apêndice A é demonstrado que o tamanho desse intervalo é da ordem de $\sqrt{n}$.

A complexidade de tempo para avaliar k polinômios de grau d utilizando o método de Horner é $O(kd)$. No caso de avaliar $\sqrt{n}$ polinômios Q_n, o resultado ainda seria obtido em $O(n)$. Para levar a complexidade de tempo abaixo da linearidade, será necessário utilizar técnicas mais avançadas de avaliação de polinômios.

2.2 Avaliação Polinomial em Múltiplos Pontos

Ao avaliar um mesmo polinômio k vezes é possível utilizar algumas técnicas que são capazes de abusar de contas repetidas que ocorrerão no processo.

Como demonstrado por [3], é possível avaliar n polinômios de grau n em tempo $O(n \log^2 n)$

assumindo que as operações aritméticas têm custo constante. Implicando que $\sqrt{n}$ avaliações da Equação 5 podem ser computadas em tempo $O(\sqrt{n}\log^2 n)$.

Uma das sub-rotinas desse algoritmo é a divisão, que funciona particularmente bem no domínio dos reais, mas pode não existir no contexto da aritmética modular. Para o problema descrito em [2] p é garantido ser um primo, *i.e.* para todo i menor que p, o seu inverso modular existe em módulo p. Dessa forma, a aplicação desta avaliação rápida de polinômios acontece de forma natural.

No futuro deste trabalho poderá ser demonstrado que o problema sobre módulo p' não-primo pode ser mapeado para um problema sobre módulo p primo.

2.3 Análise de Complexidade

A complexidade assintótica de tempo do algoritmo descrito pode ser definida por:

$$O(Build(n') + Eval(n') + dist(n, n')) \tag{7}$$

Para $Build(n')$ como o custo para se construir $Q_{n'}$; $Eval(n')$ representando o custo de computar a Equação 6; e $dist(n', n)$ como o custo de multiplicar os inteiros no intervalo $[n' + 1, n]$.

No algoritmo demonstrado por [3], não há necessidade de construção do polinômio $Q_{n'}$, uma vez que a avaliação é feita a partir de sua representação na forma do produtório de suas raízes, assim como encontramos na Equação 5. Logo, assumindo a utilização desse mesmo algoritmo e também que $dist(n', n)$ é da ordem de $O(\sqrt{n})$ como demonstrado no Apêndice A, pode-se substituir os valores na Equação 7 e concluir que a complexidade final de tempo é dada por:

$$O(\sqrt{n}\log^2 n) \tag{8}$$

Apesar disso, ainda pode haver espaço para melhorias da complexidade, em especial ao se considerar diferentes técnicas de avaliação polinomial em múltiplos pontos como sugerido em [1].

Referências

[1] Eramoni. *Fastest way to get factorial modulo a prime*. 2018. URL: `https://codeforces.com/blog/entry/63491`.

[2] *FACTMODP - Factorial Modulo Prime*. 2017. URL: `https://www.spoj.com/problems/FACTMODP/`.

[3] Justine Gauthier. "Fast Multipoint Evaluation On n Arbitrary Points". Diss. de mestr. SIMON FRASER UNIVERSITY, 2017.

[4] Ronald L. Graham, Donald E. Knuth e Oren Patashnik. *Concrete Mathematics*. Archive for History of Exact Sciences, 1991, pp. 225–249. ISBN: 0-201-14236-8.

[5] Fredrik Johansson. *Factorials mod n and Wilson's theorem*. 2012. URL: `https://fredrikj.net/blog/2012/03/factorials-mod-n-and-wilsons-theorem/`.

A Ordem do tamanho do intervalo de $[n' + 1, n]$ no domínio dos inteiros

Dado as atribuições iniciais

$$n' = \lfloor \sqrt{n} \rfloor^2$$
$$m = \sqrt{n'}$$

Considerando apenas o domínio dos inteiros, pode-se chegar nas seguintes inequações:

$$n \leq (m+1)^2$$
$$|[n'+1, n]| \leq |[m^2, (m+1)^2]|$$

Isso implica nas relações abaixo sobre os tamanhos dos intervalos:

$$|[m^2, (m+1)^2]| = (m+1)^2 - m^2 + 1$$
$$|[m^2, (m+1)^2]| = 2(m+1)$$
$$|[n'+1, n]| \leq 2(m+1)$$

Por fim, conclui-se que:

$$|[n'+1, n]| \leq 2(m+1)$$
$$|[n'+1, n]| \leq 2(\sqrt{n'}+1)$$
$$|[n'+1, n]| \leq 2(\sqrt{n}+1)$$
$$\therefore$$
$$|[n'+1, n]| = O(\sqrt{n})$$