NUMBER OF SUBMATRICES

THAT SUM TO K

Turma 05D | 2025/1

Eduardo Takashi Missaka, 10417877

Tiago Silveira Lopes, 10417600

Vitor Alves Pereira, 10410862

SUMÁRIO

[1. DESCRIÇÃO DO PROBLEMA 3](#_Toc199671976)

[2. SOLUÇÃO SERIAL 3](#_Toc199671977)

[3. PRIMEIRA TENTATIVA DE PARALELIZAÇÃO 3](#_Toc199671978)

[4. SOLUÇÃO PARALELA FINAL 3](#_Toc199671979)

[5. CONCLUSÃO 3](#_Toc199671980)

[6. REFERÊNCIAS 4](#_Toc199671981)

1. DESCRIÇÃO DO PROBLEMA

O problema conhecido como “Number of Submatrices That Sum to K” trata da contagem do número total de submatrizes (ou sub-regiões retangulares) dentro de uma matriz de inteiros, cuja soma dos elementos seja exatamente igual a um valor-alvo K. Dada uma matriz bidimensional com dimensões M x N contendo inteiros positivos, negativos ou zero, o objetivo é identificar e contar quantas submatrizes (contíguas em linhas e colunas) existem cujo somatório dos valores internos seja igual a K [1]. A entrada do programa consiste em uma matriz lida de um arquivo ou entrada padrão, contendo a quantidade de linhas e colunas, seguida pelos valores da matriz, e finalmente o valor K. A saída será um único número inteiro representando a quantidade de submatrizes encontradas que satisfazem essa condição. Este problema é particularmente interessante em termos computacionais pois envolve manipulação de somas parciais e otimizações para evitar uma abordagem puramente de força bruta, a qual se torna inviável para matrizes grandes devido ao crescimento quadrático ou cúbico do número de combinações possíveis [2][3].

1. SOLUÇÃO SERIAL

Partindo da solução serial, o problema será processado com o auxílio de duas funções trabalhando em conjunto: uma função para contar quantos subarrays possuem soma equivalente a K e uma função para retornar quantas submatrizes contíguas possui o somatório de elementos igual a K.

A primeira função, denominada “count\_subarrays\_with\_sum\_k”, recebe uma sequência unidimensional e o seu tamanho, e o valor de K. Por meio de um contador, realiza a contabilização de somas que equivalem a K. Em um laço de repetição aninhado, realiza as somas dos elementos da sequência. No fim, retorna o valor acumulado pelo contador.

Já a segunda função, denominada “solve”, recebe as dimensões da matriz a ser processada e o valor de K. Utiliza um contador interno para contabilizar os retornos de count\_subarrays\_with\_sum\_k, que nesta função irá receber um vetor auxiliar de tamanho N com a soma dos elementos de uma porção da matriz como parâmetro. Para isso, utiliza uma sequência aninhada de três laços de repetição. No fim, retornam a quantidade de submatrizes que somam K.

Embora retorne o resultado esperado, a eficiência deste algoritmo não é ideal; se a entrada recebida for uma matriz menor, com dimensões como 2x2, o tempo de execução fica na casa dos segundos, mas se as dimensões da matriz forem maiores, como 500x500 ou 1000x1000, o tempo de execução será muito alto comparativamente, estando em média na casa dos minutos. Os tempos de execução a seguir foram coletados com matrizes de dimensões 2x2, 10x10, 50x50, 500x500 e 1000x1000, e buscamos demonstrar a diferença do tempo de execução com base na entrada a ser processada.

|  |  |  |  |
| --- | --- | --- | --- |
| K | M | N | Solução Serial (s) |
| 1 | **2** | **2** | **0,172** |
| 1 | **10** | **10** | **0,1738** |
| 0 | **50** | **50** | **0,1837** |
| 0 | **500** | **500** | **18,8052** |
| 0 | **1000** | **1000** | **326,5810** |

Essa diferença ocorre em razão da complexidade do programa. Analisando as funções principais, temos uma complexidade média de tempo de O(n^2) para a função count\_subarrays\_with\_sum\_k e O(m^2n^2) para a função solve (complexidade de tempo do programa).

A seguir, utilizando uma estratégia paralela, buscamos uma tentativa de melhoria na eficiência deste algoritmo, utilizando as mesmas entradas realizadas nesta solução, a fim de demonstrar como a paralelização otimiza o algoritmo.

1. PRIMEIRA TENTATIVA DE PARALELIZAÇÃO

Apresentar a ideia de paralelismo e apresentar as tabelas de tempo de execução, speedup e eficiência. Apresentar os grãos aqui. Citar referências.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | M | N | Threads | | | | | | | Speedup | Eficiência |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Apresentar gráficos do speedup e eficiência.

1. SOLUÇÃO PARALELA FINAL

Discutir sobre escalabilidade forte e ajustes realizados para o programa. Demonstre especificamente a parte paralelizada. Citar referências.

1. CONCLUSÃO

Apresentar o que deu certo e o que “deu errado”, apresentando as dificuldades e desafios do problema.

1. REFERÊNCIAS

[1] - 1074. Number of Submatrices That Sum to Target. [*S. l.*]: AlgoMonster, 2024. Disponível em: https://algo.monster/liteproblems/1074. Acesso em: 28 maio 2025.

[2] - NUMBER of Submatrices That Sum to Target | Live Coding with Explanation | Leetcode - 1074. [*S. l.*]: Algorithms Made Easy, 2021. Vídeo. Disponível em: https://www.youtube.com/watch?v=elADMOC\_hDI. Acesso em: 28 maio 2025.

[3] - COUNT of submatrix with sum X in a given Matrix. [*S. l.*]: Geeks For Geeks, 28 nov. 2023. Disponível em: https://www.geeksforgeeks.org/count-of-submatrix-with-sum-x-in-a-given-matrix/. Acesso em: 28 maio 2025.

[4] - NUMBER of Submatrices that Sum to Target - Leetcode 1074 - Python. [*S. l.*]: NeetCodeIO, 2024. Vídeo. Disponível em: https://youtu.be/43DRBP2DUHg. Acesso em: 28 maio 2025.