

Trabalho Prático de Matemática Discreta

Livia Delgado de Almeida Carneiro

Soma Máxima

Algoritmo de Kadane

Funcionamento O algoritmo de Kadane para encontrar o subvetor com a maior soma em um vetor tem como operação principal o seguinte loop:

```
for (i = 0; i < n; i++){
    max_atual = max_atual + v[i];

    if (max_atual < 0) {
        max_atual = 0;
        xtemp = i + 1;
    }

    if (max_atual > *max_total){
        *max_total = max_atual;
        *x = xtemp;
        *y = i;
    }
}
```

Seu funcionamento é baseado no cálculo da maior soma de elementos do subvetor até o elemento atual, para todo elemento do vetor.

Como assim? Usarei como exemplo para a explicação um vetor de 4 posições:

vetor = [-13, 5, 7, -3]

Analisando sem o algoritmo, conseguimos perceber que a maior soma se encontra entre os elementos 1 e 2 (12).

Como o algoritmo calcula isso? A tabela abaixo representa o estado de cada variável a cada iteração:

iteração	max_atual	max_total	índice inicial (x)	índice final (y)
i = 0	0	0	1	-
i = 1	5	5	1	1
i = 2	12	12	1	2
i = 3	9	12	1	2

Antes da primeira iteração, as variáveis são inicializadas: o máximo atual é igual a 0 (o mínimo valor permitido independente do caso) e o máximo total inicializado com o menor valor possível para o inteiro. Na primeira iteração ($i = 0$), será temporariamente atribuído à variável `max_atual` o valor dela (0) somado ao valor da posição atual do vetor (-13). Como esse valor é menor que 0, ele não é interessante - então `max_atual` volta ao estado inicial e mínimo de 0. É atribuído à variável `xtemp` o valor de `i` para a iteração seguinte - pois é onde possivelmente se encontra o primeiro elemento do subvetor com a soma máxima. O valor máximo atual (0) é maior que o valor máximo total (`INT_MIN`), o que significa que, até o momento, o subvetor analisado (a partir do próximo elemento) representa o máximo valor total.

Ou seja, a comparação é feita no elemento atual do vetor - deseja-se saber se ele deve ser somado ao subvetor analisado (caso ele acrescente ao valor, tornando o `max_atual` maior que o `max_total`) ou se deve ser analisado um outro subvetor a partir dele.

Análise de Complexidade Ambos os custos de tempo e espaço são lineares ($O(n)$), pois ocorre pelo menos e no máximo uma iteração pelo vetor em questão.

Quadrado Mágico

Magic Square Algorithm

Funcionamento A constante mágica - isto é, o valor que representa o resultado da soma de todos os elementos de uma linha, de uma coluna ou de uma diagonal, é calculada da seguinte forma: $n * (n^2 + 1) / 2$

Quadrado Mágico Ímpar O algoritmo para preencher as posições do quadrado mágico segue a seguinte função definida por duas sentenças:

$$(n/2, n-1) \text{ se } i = 0 \text{ e } j = 0 \quad (i(a-1) - 1, j(a-1) + 1)$$

São seguidas 3 condições:

1. A posição do próximo número é calculada subtraindo 1 do número da linha e somando 1 ao número da coluna. Se o número da linha resultar em -1 ele representa $n-1$, e se o número da coluna resultar em n , ele se torna 0.
2. Se a posição calculada já estiver ocupada, será subtraído 2 do número da coluna e somado 1 ao número da linha.
3. Se o número da linha calculada for -1 e o da coluna for n , a nova posição seria $(0, n-2)$.

Análise de Complexidade Como o algoritmo trabalha com um vetor de 2 dimensões, que sempre será percorrido e preenchido, seu custo de tempo e de espaço pode ser representado por $O(n^2)$.