

Documentação: Trabalho Prático (Soma Máxima, Quadrado Mágico), Matemática Discreta

Soma Máxima em um subvetor:

Primeiramente, o programa solicita a entrada de um número para definirmos o tamanho do nosso array de 3 a 20. Caso o usuário digite um valor fora do intervalo solicitado, o programa avisa ao usuário que o mesmo cometeu um erro e solicita a entrada de um novo valor. Após a inserção do número de elementos seguido dos elementos que comporão o array, o programa segue para a execução do seguinte algoritmo:

- Passo 01) O programa percorre o array uma vez contabilizando o número de elementos positivos e negativos. Caso tenhamos só elementos positivos, a soma máxima será a soma de todos os elementos e consequentemente, o índice inicial será 1 e a final N (sendo N o tamanho inserido); caso haja apenas elementos negativos, a soma máxima será igual a 0 (por *default*) e os índices serão 0, significando que eles são “inválidos”. Se nenhuma dessas condições forem verdadeiras, o algoritmo prossegue para o passo 02.
- Passo 02) O programa passa a percorrer o array até que todos os seus “subvetores” sejam lidos. Durante a leitura, caso a soma encontrada for maior do que a já estabelecida no início (0), o valor da soma máxima é atualizado para a nova soma, juntamente com os respectivos índices inicial e final.

Ao final de tudo, o programa exibe como saída a soma máxima e os índices inicial e final do “subvetor”. A seguir, um exemplo da execução do mesmo:

Entrada:

8
9 0 0 0 3 1 1 1

Saida:

Soma máxima: 15
Índices: 1 a 8

Quadrado Mágico:

Primeiramente, o programa solicita a entrada de um número para definirmos o tamanho do quadrado mágico a ser gerado de 3 a 5. Caso o usuário digite um valor fora do intervalo solicitado, o programa avisa ao usuário que o mesmo cometeu um erro e solicita a entrada de um novo valor. Após a inserção do tamanho, o programa executa o seguinte algoritmo:

- Passo 01) Verificar se o tamanho inserido corresponde a um número par ou ímpar. Caso seja o primeiro, o programa executa o passo 02, caso contrário, o passo 03.
- Passo 02) Como no intervalo de entrada, o único valor par remete ao quadrado de lado 4 (par duplo, divisível por 2 e 4), o algoritmo foi simplificado para atender a essa condição apenas. Primeiramente, definimos uma variável “num” que inicialmente possui valor 1 e a cada posição avançada na matriz, incrementaremos seu valor. Percorreremos a matriz relacionada ao quadrado desde a posição a(1,1) até a posição a(4,4) com a seguinte condição envolvida: “se essa posição pertencer a diagonal principal ($i == j$) ou secundária ($i == N - 1 - j$, sendo N o tamanho inserido), colocaremos o valor atual de “num” naquela posição”. Quando chegarmos na posição a(4,4), o valor de ‘num’ volta a ser 1 (sendo incrementado da mesma forma anterior) e dessa vez, percorreremos a matriz começando da posição a(4,4) até

a posição $a(1,1)$, com a seguinte condição: “se não houver número na posição atual, coloque o valor atual de ‘num’”. Finalmente, trocamos a última linha com a antepenúltima e a última coluna com a antepenúltima, nessa ordem. Essa troca final é opcional, visto que já tínhamos um quadrado mágico em mãos, porém, após essa troca, teremos um quadrado mágico 4x4 chamado de *Panmagic Square*, que possui o número máximo de ocorrências da constante mágica (52 maneiras distintas de conseguir a constante mágica).

- Passo 03) Usaremos uma variável “num” que inicialmente assumirá valor 1 e, a cada iteração, terá seu valor incrementado. Para completar o quadrado mágico, usaremos um método conhecido como *Siamese Method*, que consiste em: primeiramente, inserirmos o número 1 (valor esse que é igual a nossa variável “num”) na primeira linha e na coluna central da nossa matriz. Após feito isso, percorreremos a mesma deslocando uma linha para cima e uma coluna para a direita. Se sairmos das bordas do quadrado mágico, faremos um movimento de “wrap” dando basicamente uma volta nele. Verificamos se já existe um número naquela posição. Se sim, volte para a posição anterior, desça uma linha e insira o novo número, caso contrário, apenas insira o novo número depois desse deslocamento. Verificamos se o quadrado mágico já foi preenchido por completo e, se isso não for verdade, continuamos com esse algoritmo até essa condição for verdadeira. Para o deslocamento das linhas e colunas, o programa basicamente incrementa a posição da coluna atual (j) e decrementa a da linha atual (i), e para o movimento de “wrap”, fazemos com que a nossa coluna atual passe a ser a primeira (caso nós ultrapassarmos o limite das colunas) e que a nossa linha atual passe a ser a última (caso nós ultrapassarmos o limite das linhas).

Ao final de tudo, o programa exibe como saída o valor do tamanho do quadrado, da constante mágica calculada através da fórmula fechada $(N * (N^2 + 1))/2$ (sendo N o tamanho), e o inicial quadrado mágico de tamanho N. A seguir, um exemplo da execução do mesmo:

Entrada:

4

Saida:

n = 4, Soma = 34

1 15 4 14

12 6 9 7

13 3 16 2

8 10 5 11