Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Matemática Discreta (2017.2)
Isadora Alves de Salles

Soma Máxima

Para solucionar o problema proposto, com o intuito de não percorrer o vetor inúmeras vezes e encontrar a maior soma através de comparações com todas as possíveis somas dentro do vetor, optei por um método no qual o vetor é lido apenas uma vez. Com apenas um loop é possível escrever os valores no vetor e encontrar a soma máxima deste vetor. O funcionamento é o seguinte:

Basicamente, parti do princípio de que deve-se avaliar se somar um negativo compensa ou não. Assim, se a soma dos números anteriores ao negativo for maior que o número negativo em módulo, podemos "passar" por este número negativo e continuar a soma com os números seguintes. Portanto, para que a solução seja efetiva cada soma deve ser comparada com a maior soma, que inicialmente é zero, se a soma atual for mais que a soma guardada como maior anteriormente, a maior soma se torna a soma atual.

Dentro de um loop que será repetido N vezes, sendo N o tamanho do vetor, após ler a entrada do primeiro número, este é identificado como sendo negativo ou não. Se o primeiro número for negativo, através de uma comparação em que a soma anterior somada ao número deve permanecer positiva para que o número seja somado, percebemos que o número não será adicionado, já que inicialmente temos a soma igual a 0. Isso é algo intuitivo, se estamos procurando um sub-vetor que tem maior soma dentro de um vetor, este sub-vetor não vai começar com números negativos. Se o primeiro número for positivo ele será armazenado a soma e se tornará maior, assim quando o próximo número for avaliado e este for negativo já teremos salvo a maior soma até então.

Por exemplo, tem-se o vetor 31 -30 32, avaliando inicialmente a maior soma se torna 31, depois percebemos que compensa continuar com índice 1 como inicial e passar pelo valor negativo, já que ainda teremos uma soma igual a 1, no final deste loop a maior soma ainda é 31, porém ao ler o terceiro número somamos ele à soma atual, nos dando um total de 33 que é maior que 31, então este passa a ser o valor da soma máxima.

Se no meio do vetor encontrarmos um número negativo pelo qual não compensa "passar", a maior soma está armazenada como a soma anterior a este, e agora a soma atual é zerada, e o índice inicial passa a indicar o próximo número do vetor, já que logicamente não começaria somando um negativo, e o processo recomeça. Se essa soma em algum momento se tornar maior que soma antes do negativo, essa se torna a maior soma.

O índice de fim é marcado pelo incremento da variável toda vez que mais um número é somado.

Tendo como exemplo o vetor colocado no roteiro do trabalho:

31 -41 59 26 -53 58 97 -93 -23 84

Inicialmente a maior soma é 31, após ler o segundo número percebe-se que não compensa passar por ele então, a soma atual é zerada e lê-se o terceiro número, soma atual recebe 59 e índice inicial se torna 3. Como 59>31, maior soma recebe 59. Assim após ler 26, maior soma se torna 85. Como o próximo número é negativo, analisa-se que compensa passar por ele, neste caso sim pois 85-53>0, então a soma atual é atualizada para 32, mas a maior soma continua sendo 85. Após ler 58 e 97, a soma atual recebe 187 e a maior soma neste momento também é 187. Continuamos lendo o vetor e percebemos que é possível passar pelos próximos números negativos, e passamos, porém, ao final do vetor temos soma igual a 155 que não é maior que 187, que é a maior soma guardada. Então, a maior soma é do índice 3 ao 7, sendo igual a 187.

Quadrado Mágico

Buscando solucionar o problema do quadrado mágico tentei encontrar uma maneira única de resolver tanto para os pares quanto para os ímpares, porém não encontrei, sendo assim divide meu problema em dois casos, o que resolve os ímpares (3x3 e 5x5) e o que resolve os pares (4x4).

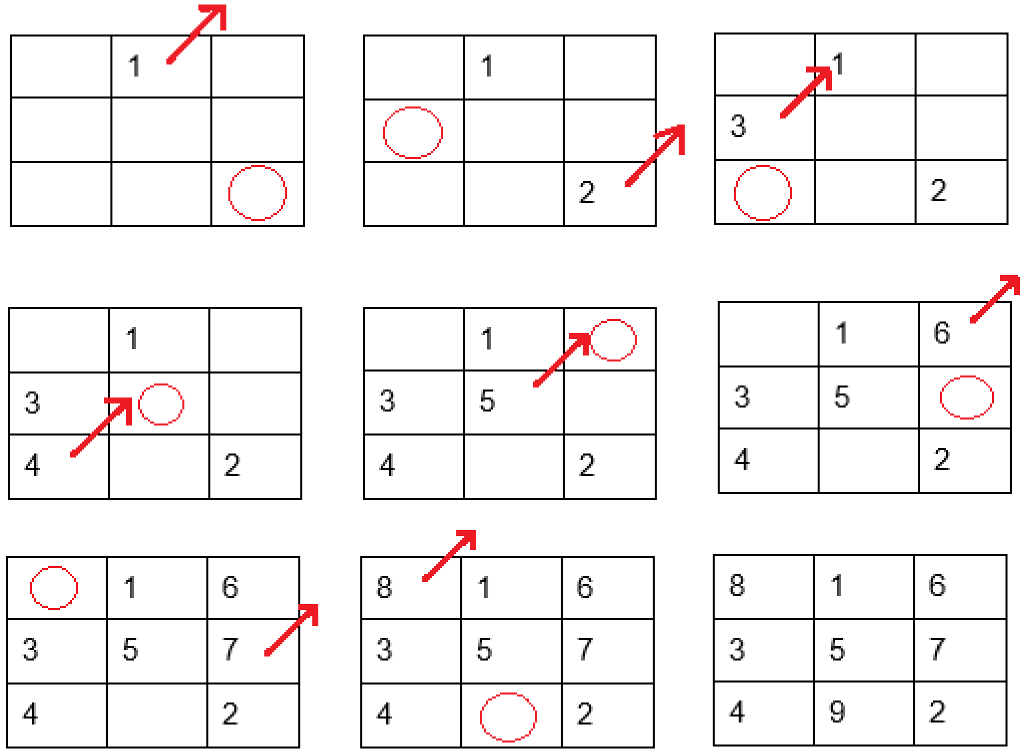
Primeiramente, para ambos os casos, após ter o número correspondente ao lado do quadrado usa-se a fórmula Soma = [n*(n*n + 1)] /2 para descobrir a qual a soma de cada linha, coluna e diagonal do quadrado. Esta fórmula é válida para todo quadrado mágico formado de números de 1 a n^2.

Agora para a formação de quadrado magico ímpar, cria-se uma matriz[n][n], e zera todas as posições dessa matriz, depois disso inicia-se com o número 1 na primeira linha e coluna do meio da matriz, ou seja, matriz[0][n/2]. A partir disso o problema é solucionado seguindo algumas regras:

- Os próximos dígitos serão adicionados em ordem, a partir do 2, sempre uma linha acima e uma fileira ao lado.
- Quando o número sai para fora da quantidade de linhas da matriz, ou seja, passa para a linha “-1” este é colocado na última linha (n-1).

- Quando o número sai para fora da quantidade de colunas da matriz, ou seja, passa para a coluna “n” este é colocado na primeira coluna (0).
- Quando a posição em que se deve colocar o número está ocupada, ele deve ser colocado abaixo do número anterior.

Seguindo estas regras conseguimos solucionar os quadrados ímpares, veja o exemplo de formação de um quadrado 3x3:



Este método funciona muito bem para quadrados ímpares, porém para quadrados pares o raciocínio não segue o mesmo e é um pouco mais difícil encontrar uma semelhança entre os quadrados pares.

Então para o quadrado 4x4 eu encontrei uma técnica em que os números são escritos de acordo com a posição, quando presentes nas diagonais, por exemplo posição 1 está na diagonal, escreve-se 1 na mesma, ao final disso teríamos as diagonais formadas e colocaríamos os números restantes, de baixo para cima na matriz. Analisando este método eu percebi que é simplesmente mais fácil gerar um quadrado 4x4 preenchendo o mesmo de 1 a 16 em ordem, e depois trocar os valores que não fazem parte da diagonal, para que ao invés de começar “1 2 3 4”, a posição do 2 estaria o 15 que é o último número a ser colocado no método anterior. Basicamente:

Inicialmente escrevi todos os números de 1 a 16 em cada espaço do quadrado, em ordem. Depois, percebi que para resolver era necessário apenas realizar a troca de alguns números de lugar. Essa troca funciona basicamente assim:

- Inicia-se na primeira linha e troca o segundo número com o terceiro número da última linha.
- Na segunda linha troca-se o primeiro número com o último número da terceira linha.
- Na terceira linha troca-se o primeiro número com o último número da segunda linha.
- Na última linha troca-se o segundo número com o terceiro número da primeira linha.

Seguindo essa ordem eu percebi que eu tinha que realizar um loop que percorre as linhas da matriz de cima para baixo, enquanto que a variável que estava sendo trocada estava sendo indexada por um índice qualquer que começava de baixo para cima na matriz, ou seja de 3 a 0.

Assim seguindo algumas condições para alterar a coluna referente aos números que queremos trocar, uma hora incrementa outra hora decrementa, consegui implementar de uma forma eficiente e produzir o quadrado mágico 4x4. Abaixo vemos a resolução deste problema:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16