Alfredo's MAC0110 Journal

Alfredo Goldman

June 14, 2020

1 Programa do curso

1.1 Aula 22 - Ainda matrizes

Dadas duas matrizes m e n, faça uma função que devolva o produto delas (sem usar o * para matrizes).

```
function multiplica(a, b)
   dima = size(a)
   dimb = size(b)
   if dima[2] != dimb[1]
     return -1
   end
   c = zeros(dima[1], dimb[2])
   for i in 1:dima[1]
      for j in 1:dimb[2]
         for k in 1:dima[2]
            c[i, j] = c[i, j] + a[i, k] * b[k, j]
         end
      end
   end
   return c
 end
```

Uma matriz quadrada de tamanho n é um quadrado latino se em cada linha e coluna aparecem todos os valores de 1 a n. Faça uma função que dada uma matriz quadrada verifica se ela é um quadrado latino.

```
#
```

Dizemos que uma matriz inteira A nxn é uma matriz de permutação se em cada linha e em cada coluna houver n-1 elementos nulos e um único elemento igual a 1. Faça uma função que recebe uma matriz quadrada e que verifica se ela é uma matriz de permutação.

```
#
```

Mas, conforme a abstração que fazemos as matrizes podem representar coisas diferentes, por exemplo dada uma matriz quadrada n x n, a posição a[i][j], pode indicar se há um caminho entre a cidade i e a cidade j.

Da mesma forma, a[i][j] pode representar a distância entre a cidade i e a cidade j. Com isso, podemos ter problemas mais sofisticados como saber se dá para chegar da cidade i na cidade j, ou o custo do menor caminho.

Nessa aula, vamos ver, um exemplo mais elaborado do uso de matriz, através de uma teoria conhecida como percolation. Vamos primeiro entender o que seria isso usando o livro do Sedgewick e do Wayne `https://introcs.cs.princeton.edu/java/24percolation/`

A pergunta é saber qual é a probabilidade, a partir da qual a percolation ocorre com grande chance (digamos mais de 80%, ou seja em 100 tentativas, em 80 ocorre a percolation). Vamos para simplificar o problema pensar em matrizes 20 por 20.

Como resolver esse problema?