# DCA0214.1 - LABORATÓRIO DE ESTRUTURAS DE DADOS
# Aula 13: Algoritmos em grafos

Prof. Felipe Fernandes

30 Maio de 2019

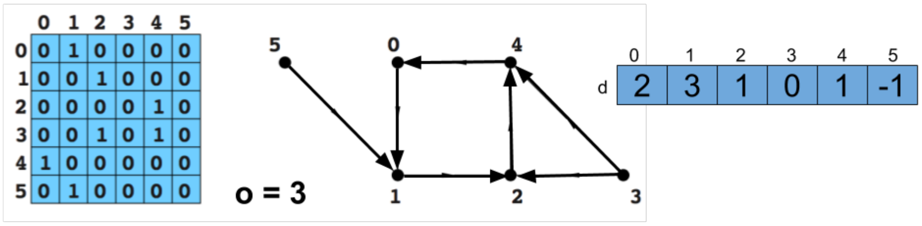
1. Uma indústria possui um estoque de $n > 1$ reagentes químicos. Por segurança, alguns pares de reagentes devem ficar separados. A fim de separar os reagentes, a indústria dispõe de dois grandes galpões. Dois reagentes quaisquer podem ficar no mesmo galpão se, e somente se, eles não representam perigo ao ficarem juntos (ou seja, são compatíveis). A indústria lhe contratou para ajudá-la na separação dos reagentes nos dois galpões.

   (a) Modele o problema da compatibilidade dos reagentes como um grafo não-direcionado.

   (b) Implemente um algoritmo eficiente que, dada a configuração de compatibilidade dos reagentes, decide se é possível separá-los em dois galpões de modo a satisfazer as restrições acima.

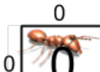
2. Seja $G(V, E)$ um grafo não direcionado, com $|V| = n$ e $|E| = m$. Faça um algoritmo, com complexidade $O(n + m)$, que verifique se existe um ciclo em $G$.

3. Suponha que temos $n$ cidades numeradas de 0 a $n - 1$ e interligadas por estradas de mão única. As ligações entre as cidades são representadas por uma matriz $A$ definida da seguinte forma: $A[x][y]$ vale 1 se existe estrada da cidade $x$ para a cidade $y$ e vale 0 em caso contrário. A figura abaixo ilustra um exemplo. Observe que, pela definição acima, **não** há garantias que $A[x][y] = A[y][x]$, pata todo $x, y$. O problema que queremos resolver é o seguinte: determinar a menor distância de uma dada cidade $o$ a cada uma das outras cidades da rede. As distâncias são armazenadas em um vetor $d$ de tal modo que $d[x]$ seja a menor distância de $o$ a $x$. Se for impossível chegar de $o$ a $x$, podemos dizer que $d[x]$ vale $-1$. Implemente um algoritmo, com complexidade $O(n^2)$, que, dado a matriz $A$ e uma cidade $0 \leq o < n$, retorne o vetor $d$ de menores distâncias.




4. Seja $G(V, E)$ um grafo não direcionado, com $|V| = n$ e $|E| = m$. Faça um algoritmo, com complexidade $O(n + m)$, que verifique se $G$ é conexo.

5. Seja um tabuleiro com $n$-por-$n$ posições, modelado por uma matriz $A[n][n]$. As posições "livres" são marcadas com 0 e as posições "bloqueadas" são marcadas com $-1$. As posições (0, 0) e (n-1, n-1) estão livres. Escreva um algoritmo $O(n^2)$ que ajude uma formiguinha, que está inicialmente na posição (0, 0), a chegar à posição (n-1, n-1). Em cada posição, a formiguinha só pode se deslocar para uma posição livre que esteja à direita, à esquerda, acima ou abaixo da posição corrente. Seu algoritmo deve imprimir o caminho a ser percorrido pela formiguinha até o destino.



| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | -1 | 0 | 0 | 0 | -1 |
| 1 | 0 | 0 | -1 | 0 | -1 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | -1 |
| 3 | 0 | -1 | -1 | 0 | 0 | 0 |
| 4 | -1 | 0 | 0 | 0 | -1 | -1 |
| 5 | 0 | 0 | -1 | 0 | 0 | 0 |