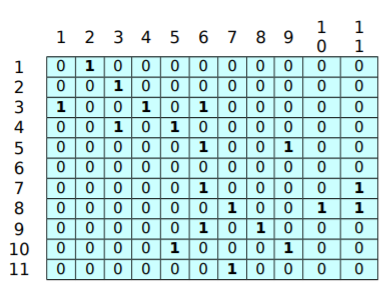
**SIN110 Algoritmos e Grafos - Exercício E4**

**Aluna:** Caroline Lopes Resek

**Matrícula:** 2017010113

1. Considere o dígrafo G2 de ordem 11, armazenado na matriz de adjacências abaixo:



Dígrafo G2:



* 1. Empregando a busca em profundidade, demonstre a sequência de visitas, e faça também um desenho da floresta de busca em profundidade.

\*Busca de profundidade:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vértice | cor(u) | Predecessor(u) | d(u) | f(u) |
| 1 | B/C/P | - | 1 | 22 |
| 2 | B/C/P | 1 | 2 | 21 |
| 3 | B/C/P | 2 | 3 | 20 |
| 4 | B/C/P | 3 | 4 | 19 |
| 5 | B/C/P | 4 | 5 | 18 |
| 6 | B/C/P | 7 | 11 | 12 |
| 7 | B/C/P | 8 | 10 | 15 |
| 8 | B/C/P | 9 | 7 | 16 |
| 9 | B/C/P | 5 | 6 | 17 |
| 10 | B/C/P | 8 | 8 | 9 |
| 11 | B/C/P | 7 | 13 | 14 |

\*Floresta em busca de profundidade:



* 1. Determine o número de componentes e, demonstre quais vértices pertencem a cada componente.

Há apenas um componente no digrafo G2, o qual é composto por todos os vértices do grafo (11) e pode ser representado pelo próprio grafo.



* 1. Existem ciclos nesse dígrafo? Mostre os ciclos.

Sim, existem 4 ciclos nesse digrafo:



1. Simule a execução da função DFS no grafo definido pelo conjunto de arestas: 0-6 0-1 0-5 1-2 2-6 6-7 7-8 7-10 10-8 5-3 5-4 4-11 4-9 4-3 9-11 11-12 (Adote a representação por listas de adjacência e insira as arestas, na ordem dada, num grafo inicialmente vazio.) Faça um desenho da arborescência de busca em profundidade do grafo.

Grafo:



Lista de Adjacência:

Busca em profundidade:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vértice | cor(u) | Predecessor(u) | d(u) | f(u) |
| 0 | B/C/P | - | 1 | 26 |
| 1 | B/C/P | 0 | 10 | 13 |
| 2 | B/C/P | 1 | 11 | 12 |
| 3 | B/C/P | 5 | 15 | 16 |
| 4 | B/C/P | 5 | 17 | 24 |
| 5 | B/C/P | 0 | 14 | 25 |
| 6 | B/C/P | 0 | 2 | 9 |
| 7 | B/C/P | 6 | 3 | 8 |
| 8 | B/C/P | 7 | 4 | 5 |
| 9 | B/C/P | 4 | 22 | 23 |
| 10 | B/C/P | 7 | 6 | 7 |
| 11 | B/C/P | 4 | 18 | 21 |
| 12 | B/C/P | 11 | 19 | 20 |

Arborescência:



1. Considere um campeonato de futebol com n times se enfrentando todos contra todos (com uma só partida entre cada par de times), onde nunca há empates (se necessário disputam-se pênaltis). Queremos saber se esse campeonato é consistente, isto é: caso um time i tenha perdido para um time j, nunca ocorrerá uma seqüência de times ti, t1, t2, ..., tk, tj tal que ti venceu t1, t1 venceu t2, ... e, tk venceu tj. Modele o problema de verificar se “o campeonato é consistente”, como um problema em grafos e proponha um algoritmo de tempo linear no tamanho do grafo para resolvê-lo.

Campeonato de futebol: não pode ter empate e pode haver apenas uma partida entre cada par de times.

Para resolver esse problema modelei um grafo de um campeonato com 4 times participantes:



Com isso, é possível perceber que o campeonato é persistente, pois todos jogam contra todos e não há empates.

Resultados dos jogos:



\*Algoritmo:

ValidaCampeonato (Times,n)

t<-Tmes

para cada i<-1 até n-1 faça

se (jaJogou(adjs(t), t) = 0))

Resultados(i) <- partidas(t)

devolve encontrarVencedor(Resultados)

O algoritmo recebe o grafo(Times) e a quantidade de times(n) como parâmetro e, para cada vértice do grafo, verifica se o vértice atual já jogou com todos os seus adjacentes senão jogou chama-se a função partida que verificará com quais adjacentes falta o vértice atual jogar e são realizadas as partidas. São retornados os resultados da partida que são armazenados em um índice do vetor Resultados. No final é retornado o vencedor do campeonato através da função encontrarVencedor. Esse algoritmo garante que um time jogará com os outros apenas uma vez e que não haverá empates, ou seja, sempre haverá um vencedor.