

# Cenário 01

## Grafos

Alunos : Gabriel Vieira Soriano Aderaldo - 1710538

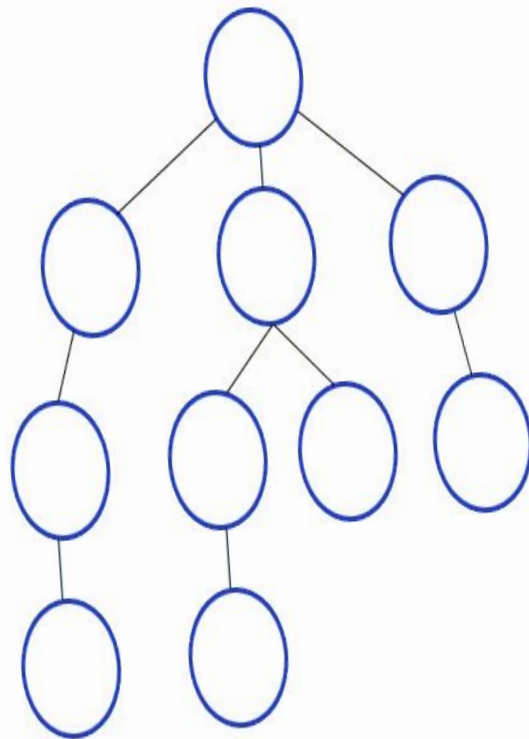
Estudos sobre **busca em largura - BFS**

[ BFS ]

Complexidade:  
 $O(|E|+|V|)$

Ano de  
publicação:  
1972

Pessoa que  
publicou:  
Konrad Zuse



Complexidade:  
 $O(|E|+|V|)$

Ano de  
publicação:  
1972

Pessoa que  
publicou:  
Konrad Zuse

## Vantagens

- Simplicidade e Facilidade de Implementação
- Encontrar o Caminho Mais Curto

BFS

Complexidade:  
 $O(|E|+|V|)$

Ano de  
publicação:  
1972

Pessoa que  
publicou:  
Konrad Zuse

## Desvantagens

- **Consumo de Memória**
- Ineficiente para Grandes Profundidades
- Complexidade de Tempo

BFS

Complexidade:  
 $O(|E|+|V|)$

Ano de  
publicação:  
1972

Pessoa que  
publicou:  
Konrad Zuse

## PSEUDO-CÓDIGO

```
BuscaEmLargura
  escolha uma raiz s de G
  marque s
  insira s em F
  enquanto F não está vazia faça
    seja v o primeiro vértice de F
    para cada w ∈ listaDeAdjacência de v faça
      se w não está marcado então
        visite aresta entre v e w
        marque w
        insira w em F
    senao se w ∈ F então
      visite aresta entre v e w
  fim se
  fim para
  retira v de F
fim enquanto
```

# Problema do cenário

Lucas é um estudante do 1º semestre de computação e gosta muito de videogames. Animado com o início de seus estudos na faculdade, ele decidiu implementar um jogo bastante conhecido na cultura geek, o pac-man. Como ele havia feito um curso de jogos, a criação da interface não foi um problema. Contudo, Lucas teve dificuldade para implementar a lógica de perseguição dos fantasmas. No jogo original, ele percebeu que alguns fantasmas possuíam comportamentos diferentes, mas um deles, o vermelho, sempre parecia buscar o menor caminho dentro do labirinto até o personagem. Você é um(a) colega de um semestre mais avançado do curso e pode ajudar Lucas na criação da lógica de perseguição do fantasma vermelho. Neste caso, indique para ele a solução mais simples possível, explicando como resolver o problema computacionalmente e demonstrando o passo-a-passo de execução do algoritmo.



# Implementação em DART

```
bool bfs({required Map<String,List<String>> graph, required init, required target}){  
  List queue = [];  
  List queue_visited = [];  
  
  queue_visited.add(init);  
  
  if(graph[init] != null){  
    graph[init]!.forEach((key){  
      queue.add(key);  
    });  
  }  
  
  while(queue.isNotEmpty){  
    final whoIsCheck = queue.first;  
    queue_visited.add(whoIsCheck);  
    queue.remove(whoIsCheck);  
  
    if(whoIsCheck == target || !queue_visited.contains(whoIsCheck)){  
      return true;  
    }else{  
      graph[whoIsCheck]?.forEach((key)=>queue.add(key));  
    }  
  }  
  return false;  
}
```

# Cenário 01

## Grafos

Alunos : Gabriel Vieira Soriano Aderaldo - 1710538

Estudos sobre **busca em largura - BFS**

[ BFS ]