



Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática
Algoritmos e Estruturas de Dados

Relatório do Projeto 2

Alunos:
Gabriel Marta - 120155
Tiago Pita - 120152

Janeiro
2025

Conteúdo

1	Objetivo	1
2	Algoritmo de Bellman-Ford	3
3	Algoritmo TRANSITIVE-CLOSURE	6

1 Objetivo

Desenvolver algoritmos sobre grafos sem pesos associados aos arcos, e efetuar a análise da eficiência computacional de algumas das estratégias desenvolvidas.

1. **TAD GRAPH:** desenvolver a função que, dado um grafo orientado, constrói o correspondente **grafo orientado transposto**.
2. **Módulo BELLMAN-FORD:** desenvolver a função que, dado um grafo, sem pesos associados às arestas, e um vértice inicial, constrói a **árvore dos caminhos mais curtos** entre esse vértice inicial e cada um dos outros vértices alcançáveis, usando o **algoritmo de Bellman-Ford**.
3. **Módulo TRANSITIVE-CLOSURE:** desenvolver a função que permite, dado um grafo orientado, sem pesos associados aos arcos, construir o grafo orientado que é o seu **fecho transitivo**. Esse grafo tem os mesmos vértices que o grafo original, existindo um arco orientado entre os vértices u e v se, no grafo original, v for alcançável a partir de u , i.e., existe um caminho orientado entre esses vértices. Os **vértices alcançáveis**, a partir de um dado vértice de um grafo orientado, deverão ser determinados usando o **módulo BELLMAN-FORD**.
4. **Módulo ALL-PAIRS-SHORTEST-DISTANCES:** desenvolver as funcionalidades que permitam, dado um grafo orientado, sem pesos associados aos arcos, construir a **matriz de distâncias** que, para cada **par de vértices**, contém a distância associada ao correspondente **caminho mais curto**, caso exista. Os caminhos mais curtos deverão ser determinados usando o **módulo BELLMAN-FORD**.
5. **Módulo ECCENTRICITY-MEASURES:** desenvolver as funcionalidades que permitam, dado um grafo orientado, sem pesos associados aos arcos, calcular características do grafo e dos seus vértices, que são baseadas nos valores de distância associados a caminhos mais curtos. Assim devem ser determinados: 1) **a excentricidade de cada vértice** de um grafo (i.e., a maior distância entre esse vértice e cada um dos outros vértices do grafo), 2) **o raio de um grafo** (i.e., o menor valor de excentricidade de todos os seus vértices), 3) **o diâmetro de um grafo** (i.e., o maior valor de excentricidade de todos os seus vértices), e 4) **o conjunto dos vértices centrais** de um grafo (i.e., o conjunto dos vértices cuja valor de excentricidade é igual ao raio do

grafo). As distâncias entre pares de vértices deverão ser determinadas usando o **módulo ALL-PAIRS-SHORTEST-DISTANCES**.

6. **Análise da Complexidade:** Caracterizar a complexidade algorítmica das soluções implementadas para 1) o **algoritmo de Bellman-Ford**, e para 2) o **algoritmo de construção do fecho transitivo** de um grafo.

2 Algoritmo de Bellman-Ford

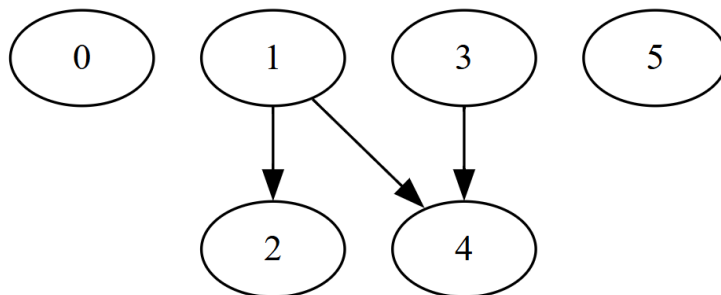
O algoritmo de Bellman-Ford é utilizado para encontrar os caminhos mais curtos de um vértice de origem para todos os outros vértices em um grafo.

Complexidade do Algoritmo de Bellman-Ford implementado:

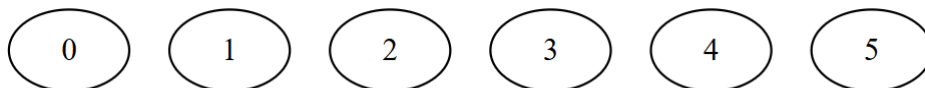
A complexidade do algoritmo de Bellman-Ford implementado é $O(V * E)$, onde V é o número de vértices e E é o número de arestas no grafo. Isso ocorre porque o algoritmo realiza relaxamento das arestas $|V| - 1$ vezes, e em cada iteração, percorre todas as arestas.

Output experimental:

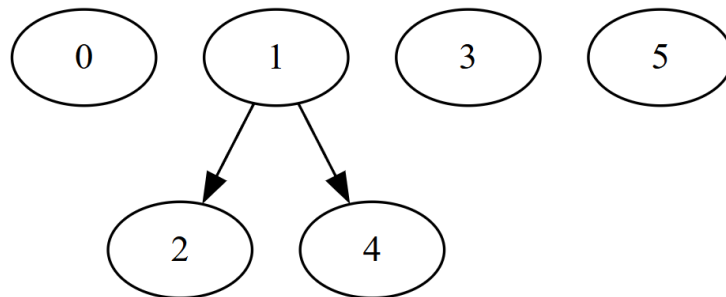
Grafo original



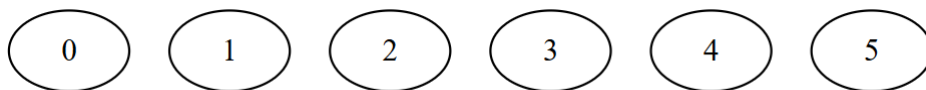
The shortest path tree rooted at 0



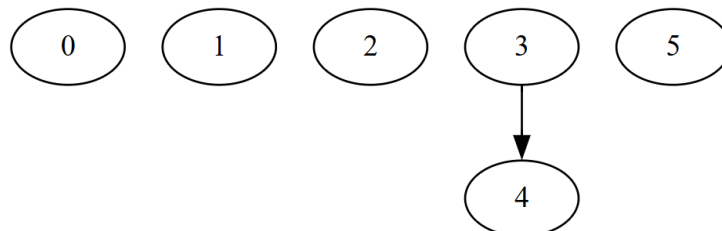
The shortest path tree rooted at 1



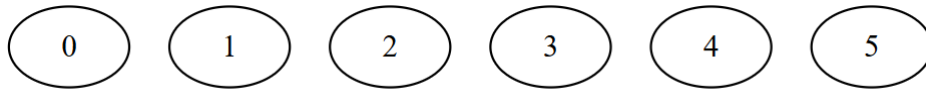
The shortest path tree rooted at 2



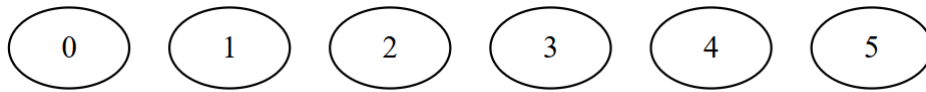
The shortest path tree rooted at 3



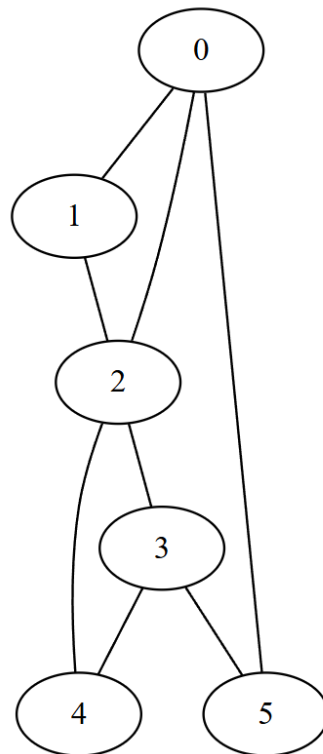
The shortest path tree rooted at 4



The shortest path tree rooted at 5



The shortest path tree rooted at 6



3 Algoritmo TRANSITIVE-CLOSURE

A função é uma implementação que calcula o fecho transitivo de um grafo direcionado. O fecho transitivo de um grafo é um novo grafo no qual existe uma aresta de um vértice u para um vértice v se e somente se existe um caminho de u para v no grafo original.

A função serve para determinar todas as possíveis conexões indiretas entre os vértices de um grafo direcionado. Isso é útil em várias aplicações, como análise de redes, onde é importante saber se existe um caminho entre dois vértices, mesmo que não haja uma conexão direta entre eles.

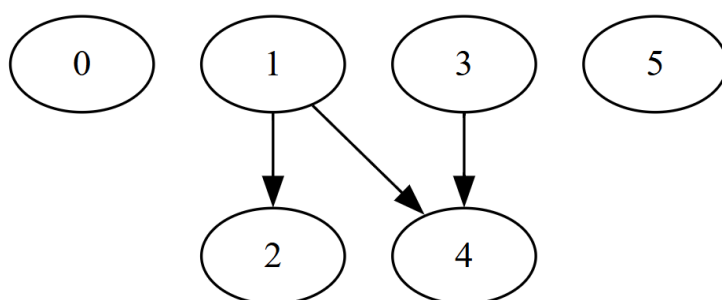
O algoritmo TRANSITIVE-CLOSURE tem complexidade $O(V^2E)$:

Um loop externo itera V vezes, chamando o algoritmo de **Bellman-Ford** ($O(VE)$) em cada iteração. Dentro desse loop, um loop interno itera também V vezes, verificando a acessibilidade em $O(V)$.

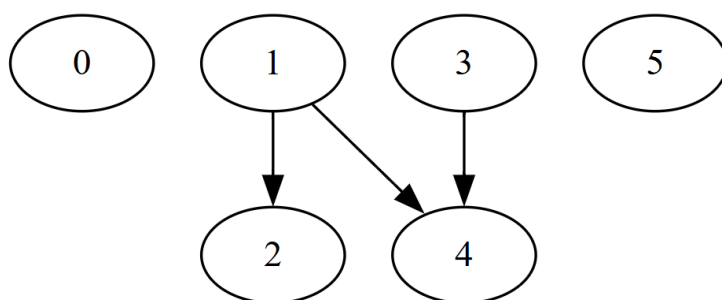
As operações *GraphAddEdge* e *GraphBellmanFordAlgDestroy* têm complexidade $O(1)$ e $O(V)$, respectivamente, não tendo impacto significativo à complexidade final, que resulta da execução do Bellman-Ford V vezes, totalizando $O(V^2E)$.

Output experimental:

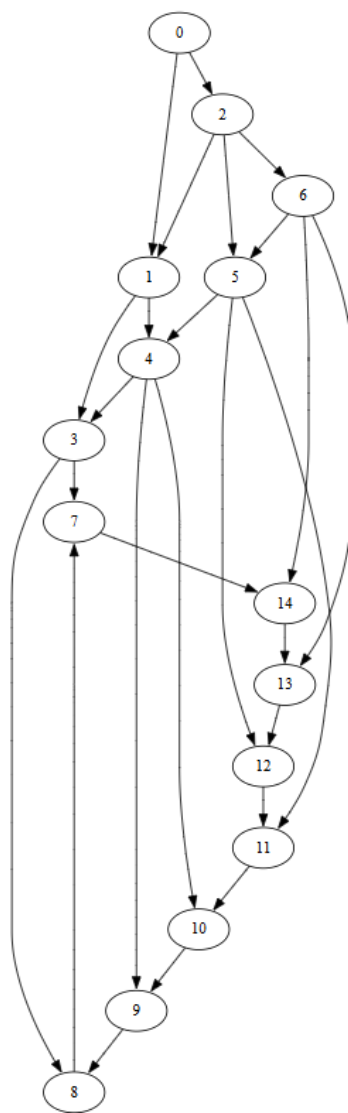
Grafo original



Grafo com fecho transitivo



Grafo original



Grafo com fecho transitivo

