



Projeto e Análise de Algoritmos Trabalho 4

Neste trabalho deverá ser realizada a análise assintótica de algoritmos referentes a grafos e redes de fluxo. O objetivo é analisar linha a linha do código fonte e encontrar o polinômio de custo do algoritmo. **Não é necessário implementá-lo e realizar experimentos práticos, apenas mapear o custo da forma mais precisa possível.**

Cada equipe ficará encarregada de analisar estratégias distintas.

O código deve ser enviado contendo todos os mapeamentos dos custos de forma a permitir verificação posterior.

As linguagens de programação que podem ser adotadas neste trabalho são C, C++, Python e Java. A escolha entre estas fica a cargo da equipe, não sendo aceitas linguagens além das citadas.

Para cada algoritmo analisado a equipe deverá identificar e explicar quanto ocorre os melhores e piores casos de execução daquele algoritmo.

Entrega: 05/06/2024 até as 12:00

Grupo 1 - Guilherme Marchiotti Ferraz e Roberval Requião Junior

- Algoritmo de Dijkstra sendo executado a partir de cada vértice do grafo
- Floyd-Warshall

Grupo 2 - Gabriel Norato Claro, Gabriel Alves Mazzuco e Rodrigo Brickmann Rocha

- Algoritmo de Push-Relabel

Grupo 3 - Vinicius Drage, Gustavo Martini e Gustavo Macedo

- Algoritmo de Dijkstra
- Algoritmo de Bellman-Ford

Grupo 4 - Gustavo Henrique Marchezoni Orlandini, Lucas David Tomalack de Souza e Rafael Roberto Hoffmann

- Algoritmo de Ford-Fulkerson
- Algoritmo de Edmonds-Karp

**Grupo 5 - Jaqueline Cavaller Faino, Davi Marchetti Giacomel e
Maria Eduarda Crema Carlos**

- Algoritmo de Johnson

**Grupo 6 - Heloisa Aparecida Alves, Eduarda Elger e
Ellen Carine Bonafin Marques**

- Algoritmo de Prim
- Algoritmo de Kruskal

**Grupo 7 - Gabriel Santos Ramos, Lucas Becker da
Conceição e Rafael Gotz**

- Algoritmo de Dijkstra usando heap de máximo
- Algoritmo de Dijkstra usando lista simples

**Grupo 8 - Alonso Lucca Fritz e
Lucca Abbado Neres**

- Algoritmo de Push-Relabel

**Grupo 9 - Arthur Augusto Eich e
Gustavo Pauli da Luz**

- Busca em Largura
- Busca em Profundidade