

Relatório de Desenvolvimento do Algoritmo Kosaraju

Introdução

Este relatório descreve o processo de construção do algoritmo Kosaraju para encontrar componentes fortemente conectados em um grafo, desenvolvido por quatro alunos de graduação ao longo de quatro dias, com duas horas de trabalho por dia. O desenvolvimento envolveu diversas etapas, desde a compreensão teórica do algoritmo até a implementação prática e a resolução de desafios encontrados durante o processo.

Participantes

- CARLOS ALDRIM FREIRE MELO FILHO - 499075
- FRANCISCO RENAN LIMA RODRIGUES - 494217
- JAIRO GOMES EVARISTO - 497466
- LUIS FELIPE DOMINGOS VIEIRA TORRES - 473253

Dia 1: Compreensão Teórica

Atividades

- Estudo individual: Cada aluno estudou a teoria por trás do algoritmo Kosaraju e componentes fortemente conectados.
- Discussão em grupo: Reunião para compartilhar o entendimento do algoritmo e esclarecer dúvidas.

Desafios

- A teoria era nova para nós, então precisamos de mais tempo para compreendê-la totalmente.
- Tivemos dificuldades em entender o conceito de transposição de grafos.
- Compreendemos a teoria rapidamente, mas tivemos dúvidas sobre a aplicação prática.

Dia 2: Planejamento e Estruturação

Atividades

- Divisão de tarefas: Planejamento das etapas de implementação e atribuição de tarefas.
- Estruturação inicial: Definição das funções principais e pseudocódigo.

Desafios

- Tivemos dificuldades em adaptar a teoria para o código.
- Não estávamos familiarizado com a estruturação de grafos em Python.
- Enfrentamos desafios ao planejar a transposição do grafo.

Dia 3: Implementação

Atividades

- Codificação das funções: Implementação das funções `read_graph`, `dfs`, `transpose_graph` e `kosaraju`.
- Teste e depuração: Teste inicial do código e correção de erros.

Desafios

- Tivemos dificuldade em implementar a função `dfs`.
- Cometemos erros ao manipular listas de adjacência.
- Enfrentamos dificuldades na transposição do grafo.
- Tivemos problemas com a lógica de processamento dos vértices na ordem inversa.

Dia 4: Testes e Finalização

Atividades

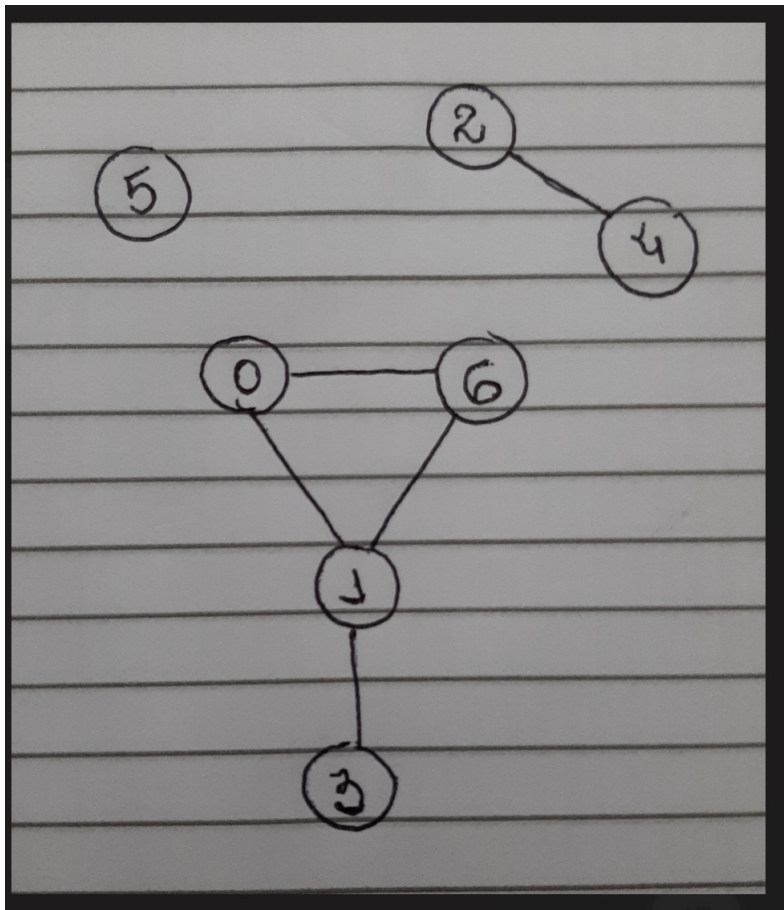
- Teste final: Teste do algoritmo com diferentes grafos.
- Correções: Ajustes finais e melhorias no código.
- Documentação: Elaboração de comentários e documentação do código.

Desafios

- Tivemos dificuldade em depurar erros lógicos.
- Enfrentamos desafios na manipulação de listas de adjacência durante os testes.
- Tivemos dificuldades em garantir a eficiência do código.

Resultados

Ao final do trabalho, conseguimos implementar o algoritmo Kosaraju com sucesso. O código passou nos casos que testamos e o processo foi documentado adequadamente.



```
python3 main.py                                o/grafos_algoritmo 21:14
Componente 1: 5
Componente 2: 2 4
Componente 3: 0 1 3 6
clear                                           o/grafos_algoritmo py || 3.11 18% ↓ 22:26
```

Conclusão

A implementação do algoritmo de Kosaraju para encontrar componentes fortemente conectados em um grafo dirigido demonstra uma abordagem eficiente e estruturada para resolver esse problema fundamental em teoria dos grafos.