

---

Aluno: Matrícula: Valor: 25,0 Nota:

---

## Trabalho Prático II

### 1. Objetivos.

- Reforçar o aprendizado sobre os algoritmos de caminhos mínimos.
- Aprimorar a habilidade de programação de algoritmos em grafos.
- Aplicar os conhecimentos em algoritmos para resolver problemas reais.

### 2. Descrição.

A gestão adequada de projetos é essencial para que os mesmos sejam concluídos com o escopo, prazo e recursos estabelecidos. Para assegurar o cumprimento do prazo de um projeto, uma técnica muito utilizada é o método do caminho crítico. Nesse método, identificamos a sequência de etapas e dependências entre elas mais crítica para a finalização do trabalho. Aquela na qual qualquer atraso atrasa o tempo mínimo de conclusão do projeto.

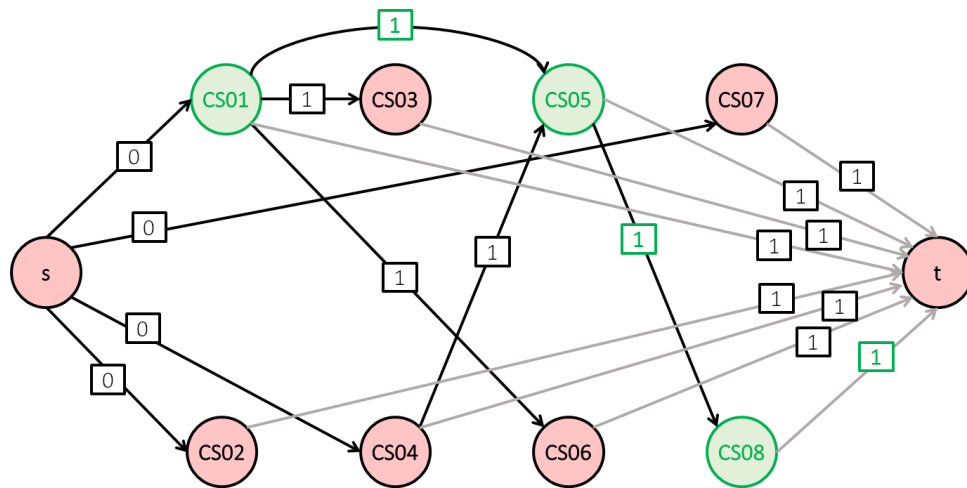
Vejamos um exemplo ligado à conclusão de um curso de graduação. Na Tabela 1, temos 8 disciplinas e a informação de dependências entre elas:

Código	Nome	Período	Duração (períodos)	Dependências
CS01	Lógica de Programação	1	1	
CS02	HTML e CSS	1	1	
CS03	Algoritmos	2	1	CS01
CS04	Orientação a Objetos	2	1	
CS05	Engenharia de Software	3	1	CS01;CS04
CS06	Banco de Dados	3	1	CS01
CS07	Redes de Computadores	4	1	
CS08	Sistemas Web e Mobile	4	1	CS05

Tabela 1: Exemplo de tarefas e dependências de um (breve) curso de graduação.

Nesse exemplo não é difícil identificar que o caminho crítico para conclusão do curso é: Lógica de Programação → Engenharia de Software → Sistemas Web e Mobile. Como cada tarefa demanda um tempo de um período, o tempo mínimo de conclusão do projeto é três períodos.

A tarefa de encontrar o caminho crítico de um projeto, dados suas etapas, duração prevista e dependências pode ser automatizada através de algoritmos em grafos. O exemplo anterior poderia ser representado como um grafo da seguinte forma:



onde cada nó representa o início de uma etapa e as arestas informam as relações de dependências. O peso de cada aresta é dado pelo tempo que a tarefa origem leva para ser concluída. Temos ainda dois nós adicionais,  $s$  indicando o início do projeto e  $t$  indicando a conclusão de tarefas. O nó  $s$  se liga diretamente a todas as tarefas sem dependências e todas as tarefas se ligam ao nó  $t$ . A caminho crítico é dado pelo caminho máximo de  $s$  a  $t$  nesse grafo, ressaltado pelos nós em verde.

O professor preparou dois arquivos de teste, ambos nos formatos .xlsx e .csv. O arquivo TOY trás o exemplo da Tabela 1; já os arquivos SJM e CJM representam os cursos de Sistemas de Informação e de Engenharia de Computação da Ufop. Esses arquivos foram criados com base na grade curricular oficial dos cursos.

Cada projeto deve:

- Ler o arquivo de dados e carregar o grafo correspondente ao arquivo (dica: a biblioteca **Pandas** oferece ótimo suporte nessa tarefa);
- Ajustar algum algoritmo de caminho mínimo para resolver o problema do caminho máximo;
- Resolver o problema, obtendo o caminho máximo (crítico) do projeto;
- Exibir para o usuário quais tarefas fazem parte do caminho crítico e qual o tempo mínimo de conclusão do projeto;
- Criar arquivos de teste representando a grade curricular de dois cursos da Ufop. O professor irá atribuir qual aluno/dupla criará os dados de quais cursos.

### 3. Interação com o usuário

A interação com o usuário deve ocorrer na função `main` do seu programa. O mesmo deve solicitar ao usuário o arquivo de entrada e, após a execução, informar o caminho máximo de  $s$  a  $t$ , bem como o tempo mínimo de conclusão do projeto (distância do caminho). Caso o caractere 0 seja informado, o programa deve encerrar. Segue um exemplo de interação com o programa:

```
Informe o arquivo (0 para sair): <critical_path/TOY.csv>

Processando...

Caminho Crítico:
- Lógica de Programação
- Engenharia de Software
- Sistemas Web e Mobile

Tempo Mínimo: 3

Informe o arquivo (0 para sair): <0>
```

### 4. Avaliação.

O trabalho deverá ser feito individualmente ou em dupla. Os alunos podem optar por estender o código do repositório da disciplina com as novas implementações, ou por iniciar um novo projeto do zero. Qualquer linguagem de programação/tecnologia pode ser usada, porém o plágio (da web e/ou de outros colegas) não é permitido. Não é requerido nenhum relatório. Apenas informe no Moodle o nome da dupla (se houver) e a URL para o projeto no GitHub. O prazo para envio do trabalho é dia 24/03/23. Caso se tenha alguma dúvida com relação à autoria do trabalho o professor poderá solicitar uma apresentação presencial ao aluno (ou dupla).

Bom trabalho!