

**SCC0216 Modelagem Computacional em Grafos**

Profa. Maria Cristina

**Projeto Prático**

**Submissão até 1 de Julho (eDisciplinas)**

Os alunos que optarem por desenvolver um projeto prático podem escolher um dos temas a seguir, e desenvolver o projeto em grupos de 2. Com base nos estudos na disciplina, os grupos podem propor e incluir outras funcionalidades mais avançadas além das sugeridas.

Os grupos devem utilizar como base suas implementações do TAD grafo (matriz de adjacências ou listas de adjacências, conforme preferência dos desenvolvedores). Os trabalhos serão avaliados de acordo com os seguintes critérios:

- atendimento aos requisitos especificados;
- desenvolvimento adicional apresentado;
- corretude das funções implementadas;
- qualidade do código-fonte (princípios de TAD, organização dos arquivos, legibilidade do código, documentação interna, complexidade do algoritmo, etc.);
- usabilidade e clareza da interface com o usuário e das saídas produzidas pelo projeto (em interface gráfica ou linha de comando, conforme preferência dos desenvolvedores).

Os grupos deverão submeter em Atividade no eDisciplinas um arquivo compactado com o código-fonte completo em C (desenvolvido em compilador de sua escolha) e relatório sucinto (de 10 páginas, aproximadamente) com as seguintes seções:

1. nome e número USP de cada membro do grupo;
2. tema escolhido para o trabalho;
3. breve introdução ao sistema desenvolvido;
4. breve explicação da implementação, citando decisões de projeto e funcionalidades implementadas;
5. demonstração de execução do sistema (com telas do sistema).

Basta que um dos desenvolvedores faça a submissão. Após a submissão, os desenvolvedores devem agendar com a professora uma data e horário para demonstração do sistema.

Data máxima para submissão: **1 de julho**

- Atenção: a apresentação pode ser agendada assim que for feita a entrega (antes desse prazo)

Para outras eventuais dúvidas sobre o projeto, consulte o quanto antes a professora ou os monitores da disciplina.

## Tema 1: Sudoku

Utilizar algoritmos em grafos para desenvolver um ‘resolvedor’ de sudoku. O programa deve resolver o desafio, completando a grade do sudoku a partir de uma grade parcialmente preenchida. O sistema deve admitir diferentes níveis de dificuldade. Compare o desempenho de sua estratégia inteligente baseada em grafos com uma estratégia simples de ‘força bruta’: faça simulações com diferentes configurações iniciais de preenchimento da grade (diferentes níveis de dificuldade), contabilizando e tabelando os tempos de execução e fazendo uma análise comparativa dos resultados obtidos com as duas estratégias.

(veja

[http://pi.math.cornell.edu/~mec/Summer2009/meerkamp/Site/Solving\\_any\\_Sudoku\\_I.html](http://pi.math.cornell.edu/~mec/Summer2009/meerkamp/Site/Solving_any_Sudoku_I.html))

O grupo deverá avaliar a melhor forma de modelar a situação com grafos e que algoritmos utilizar.

## Tema 2: Rede Social Alunos USP

Nas atuais redes sociais, a amizade é um conceito bastante amplo: na lista de contatos de cada usuário, há amigos próximos, colegas e apenas conhecidos (muitas vezes sugeridos pelo próprio aplicativo de redes sociais). Queremos criar um aplicativo de rede social voltado para os estudantes da USP São Carlos, mas que contemple somente os “amigos de verdade”.

Define-se como “amigos de verdade” aqueles que têm muita afinidade entre si, ou seja, assistem os mesmos filmes, frequentam os mesmos lugares, gostam das mesmas coisas, curtem os mesmos livros, etc.

A partir de um arquivo texto com as informações dos usuários, você deve desenvolver um sistema que modele o grafo de relacionamentos e permitir a interação dos usuários. Mostra-se abaixo um exemplo de arquivo com os dados pessoais, que podem ser definidos por você.

Exemplos de dados pessoais para montagem do grafo de relacionamentos

nome do usuário: beltrano
gênero: masculino
idade: 19
filme predileto: vingadores
local predileto: bar do zé
livro: 1984
hobby: cinema
esporte: corrida
nome do usuário: ciclano
gênero: feminino
idade: 21
filme predileto: prometheus

local predileto: praça xv livro: feliz ano velho hobby: leitura esporte: ping pong ...
--

Com base na interação com o usuário, seu sistema deve (minimamente):

- estabelecer relações de amizades entre usuários, permitindo que um usuário faça um convite de amizade a outro usuário e o outro aceite ou não, informando as chances da amizade ser verdadeira;
- sugerir possíveis novos “amigos de verdade” com base em afinidade de perfil;
- detectar pessoas na lista de contatos de um usuário que não deveriam estar lá (devido à baixa afinidade);
- identificar as pessoas com perfil ‘extrovertido, i.e., com um número de “amigos de verdade” muito acima da média, e com perfil ‘introvertido’, i.e., com um número de “amigos de verdade” muito abaixo da média.

Os grupos deverão avaliar a melhor forma de modelar a situação com grafos e quais algoritmos usar para atender as solicitações apresentadas.

### Tema 3: Sistema de Recomendação de Músicas

O **Last.fm** é um site com função de rádio online agregando uma comunidade virtual para troca de informações e recomendações sobre música. O Last.fm constrói um perfil do gosto musical de cada usuário, reunindo e exibindo suas músicas e artistas favoritos numa página feita com as informações coletadas e gravadas por um *plugin* do próprio site instalado no aplicativo de execução de música do usuário.

As músicas são associadas a *tags* que definem o gênero, por exemplo *rock*, *pop*, *country*, *jazz*, *soul*, *indie*, *hip-hop*, *mpb*, *sertanejo*, *samba*, etc. Sempre que o usuário ouve uma música, um agente (o Scrobbler) envia essa informação para a Last.fm. O *scrobbling* ajuda a saber quais músicas a pessoa ouve com mais frequência, quantas vezes ela ouviu um artista, que outros usuários têm gostos musicais parecidos (entre outras informações).

Queremos fazer um sistema similar com funcionalidades básicas. A partir de um arquivo texto com informações básicas sobre as músicas, seu sistema deve modelar o conteúdo em grafos, representando uma rede de músicas. Para cada usuário, você tem também um arquivo texto com as músicas mais ouvidas (um número arbitrário para cada usuário). Mostra-se, abaixo, um exemplo de como deve ser o arquivo de músicas e o arquivo de usuários.

Exemplo de dados sobre as músicas

nome: knocks me off my feet  
artista: steve wonder  
tags: rock pop soul

nome: change the world  
artista: eric Clapton  
tags: rock pop

nome: sabiá  
artista: antonio carlos jobim  
tags: mbp jazz bossa-nova

nome: whatching the wheels  
artista: john lennon  
tags: rock pop acoustic

...

Exemplo de dados de usuário:

username: fulano  
chega de saudade  
águas de março  
samba de verão  
...

username: beltrano  
change the world  
knocks me off my feet  
watching the wheels  
while my guitar gently weeps  
...

Com base na interação com o usuário, seu sistema deve (minimamente):

- a partir de uma música informada pelo usuário, sugerir outras músicas parecidas (em termos das tags);
- usando percurso em grafo, listar todas as músicas em uma determinada categoria (tag);
- dado um usuário, sugerir músicas que ele não ouve, mas poderia gostar;
- dado um usuário, identificar outros usuários com gostos musicais similares.

Os grupos deverão avaliar a melhor forma de modelar a situação com grafos e que algoritmos utilizar. Talvez seja necessário usar alguma medida de similaridade lexical entre textos (por exemplo, a medida do cosseno ou uma métrica de sobreposição de palavras).