

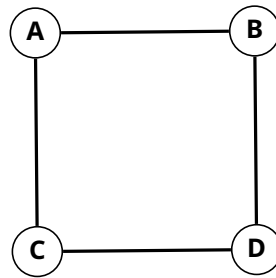
Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe e, evidentemente, a clareza. Nesta prova, você deverá sobretudo escrever códigos com complexidade assintótica menor possível.
- P: Posso fazer a lápis?
R: Não. A prova deverá ser feita a caneta.
- P: Posso responder na folha de questões?
R: Não. A prova deverá ser respondida na folha de respostas.
- P: Quanto vale esta prova?
R: Esta prova vale **30** pontos. Há mais pontos distribuídos na prova, no entanto, você poderá somar no máximo **30** pontos.

(4 pontos) 1) Explique, com suas palavras, a diferença entre um grafo dirigido (digrafo) e um grafo não dirigido. Dê um exemplo para cada.

(4 pontos) 2) O que é um ciclo em um grafo? Um grafo pode ter mais de um ciclo? Explique.

(2 pontos) 3) Represente o seguinte grafo por: Lista de adjacência, e Matriz de adjacência



(2 pontos) 4) Explique quando pode ser melhor usar listas ou matrizes de adjacência na representação de um grafo.

(2 pontos) 5) Qual a definição de grafo dirigido fortemente conexo?

(2 pontos) 6) Qual a definição de *Fecho Transitivo* de um grafo dirigido?

(4 pontos) 7) Em um grafo não ponderado, é necessário encontrar o menor número de arestas entre um vértice inicial e um vértice de destino.

1. Qual algoritmo de busca (BFS ou DFS) é mais adequado para este problema? Justifique sua escolha.
2. Implemente, em C, uma função que, dado um grafo não ponderado representado por lista de adjacência, dois vértices s (origem) e t (destino), retorna a quantidade mínima de arestas entre s e t . Se não houver caminho, retorne -1 .
3. Qual a complexidade assintótica da sua solução?

(5 pontos) 8) Implemente, em C, uma busca em profundidade (**DFS**) recursiva que, além de percorrer todos os vértices do grafo a partir de um vértice inicial, registre em dois vetores os tempos de descoberta (**pre**) e de término (**pos**) para cada vértice. Explique a utilidade desses tempos na análise ou processamento de grafos.

(5 pontos) 9) Na escola onde você trabalha, a equipe da organização da “Festa Cultural de Times Estratégicos” (FCTE) quer garantir que os alunos sejam divididos em apenas dois times para participar dos desafios. Porém, alguns alunos possuem rivalidades (foram fornecidos pares de alunos que não podem estar juntos no mesmo time, a “lista de rivalidades FCTE”).

A organização pediu sua ajuda para descobrir: **É possível formar dois times de modo que rivais nunca fiquem no mesmo grupo?**

- Explique como transformar este problema em um problema de grafos.
- Implemente, em C, um algoritmo eficiente que, dada a lista de alunos e rivalidades FCTE, decida se é possível montar os dois times conforme a regra. Justifique a escolha do algoritmo e estruturas utilizadas.

(5 pontos) 10) Na empresa em que você trabalha, há uma ferramenta de controle de permissões chamada “Fluxo Geral de Autorizações” (FGA). Trata-se de um conjunto de regras do tipo “se Fulano pode acessar X, então também pode acessar Y”.

Um analista pediu sua ajuda para responder: **dado um conjunto de permissões e relações do FGA, como obter automaticamente todas as permissões que cada usuário pode ter, levando em conta essas implicações indiretas?**

Descreva um algoritmo eficiente para resolver este problema, apresentando o raciocínio, a estrutura de dados envolvida e, em C, um esqueleto que mostre a abordagem.

(5 pontos) 11) Na *Fábrica Geral de Aplicativos* (FGA), diversos projetos precisam ser concluídos, mas alguns só podem começar depois que outros terminarem (por exemplo: “O projeto B só pode iniciar quando o A terminar”).

Um gerente te entregou uma lista com pares do tipo “Projeto X deve ser finalizado antes do Projeto Y”. Você deve determinar uma ordem válida de execução para todos os projetos, respeitando as restrições fornecidas. Se não for possível obedecer todas as dependências (por exemplo, por causa de dependências circulares), explique como detectar essa situação.

- Explique qual modelo de estrutura de dados é adequado para representar este problema.
- Mostre um algoritmo eficiente, em C, para gerar uma ordem de execução possível para os projetos, se existir.
- Dê um exemplo pequeno (com 4 projetos e 3 restrições) mostrando o resultado do seu algoritmo.