Universidade de São Paulo – USP
Instituto de Matemática e Estatística – IME
Tópicos de Programação – Verão 2025 (EAD Noturno)
Prof. M.Sc. Rodrigo Hagstrom
Segunda Avaliação
Prova Tipo 1

Estudante:
NUSP:

Instruções:

1. Cada estudante foi previamente sorteado para realizar um dos 3 tipos de prova. Verificar a lista com o resultado do sorteio disponível no ambiente de aprendizagem remota.
2. A prova tem duração 3,0 horas (19:30 às 22:30 horas).
3. A entrega deve ser realizada em um documento formato “.ODT” ou formato “.PDF”. Outros formatos serão desconsiderados.
4. Dada a natureza da disicplina ser EAD, não está proibida a consulta de materiais. Entretanto, deve-se observar as orientações constantes da prova.
4. A questão 1 deve ser resolvida em escrita manual. Sugere-se o envio de imagens da resolução. Outras formas de envio desta questão serão desconsiderados, acarretando a nota 0. para a questão em tela.
5. As demais questões podem ser resolvidas com editores de texto ou IDE de programação à sua escolha. Comentários que demonstrem que o(a) estudante sabe o que cada linha está executando são obrigatórios. Entretanto, cumpre destacar que respostas automáticas de serviços de Inteligência Artificial do tipo “Chat GPT” estão catalogadas. Cópias literais destas respostas acarretam penalidade na nota. Cópias de comentários para cada linha recebem o mesmo tratamento.
6. As respostas de provas semelhantes serão comparadas. Cópias literais das resoluções de colegas acarretam em penalidades na pontuação de questões.
7. O uso de linguagem Java ou Python é permitido, desde que sigam as mesmas instruções fornecidas em sala de aula para o uso destas linguagens nesta disciplina.
8. Cada questão vale 2,00 pontos. Consideraremos respostas parciais, nas quais a nota será parcial para a questão eventualmente com resposta incompleta.

Questão 1 (Valor na prova 35% - 3,5 pontos) - O **algoritmo de Bellman-Ford** é utilizado para encontrar o menor caminho a partir de um vértice de origem para todos os outros vértices em um grafo ponderado, mesmo quando há pesos negativos.

Enunciado:

Considere um grafo direcionado e ponderado, representado pela seguinte tabela de arestas:

Origem	Destino	Peso
A	B	4
A	C	2
B	C	3
B	D	2
C	B	1
C	D	5
D	E	-3
E	C	-2

O vértice de origem é **A**.

1. Aplique o **algoritmo de Bellman-Ford** para encontrar as menores distâncias de **A** para todos os outros vértices.
2. Mostre passo a passo a atualização das distâncias após cada iteração.
3. Explique como o algoritmo trata pesos negativos e identifique se há um ciclo de peso negativo no grafo.

Questão 2 – Assinale V (verdadeiro) ou F (falso) para as 20 assertivas abaixo. Para cada item marcado como falso você deve justificar a sua resposta. Para os itens marcados como verdadeiros você deverá escolher ao menos 4 e provar a assertiva. Método de Pontuação da Questão: marcação com V ou F (35%), justificativas e provas (65%) - Pontuação Total na Avaliação (65% - 6,5 pontos)

Problema do Caixeiro Viajante (TSP)

1. ( ) O Problema do Caixeiro Viajante pertence à classe NP-Hard e não possui uma solução polinomial conhecida.
2. ( ) Um algoritmo guloso sempre encontra a solução ótima para o TSP em grafos completos.
3. ( ) O TSP pode ser resolvido por força bruta em O(n!), testando todas as permutações possíveis.
4. ( ) O algoritmo Branch and Bound pode ser utilizado para reduzir o espaço de busca do TSP.

Algoritmo de Dijkstra

5. ( ) O Algoritmo de Dijkstra funciona corretamente mesmo que existam pesos negativos nas arestas.
6. ( ) O Algoritmo de Dijkstra encontra o caminho mínimo de um único vértice-fonte para todos os outros vértices.
7. ( ) O pior caso do Algoritmo de Dijkstra com uma fila de prioridade baseada em heap binário é O((V+E)logV).
8. ( ) O Algoritmo de Dijkstra pode ser implementado eficientemente usando uma fila de prioridade (Heap).

Busca em Largura (BFS) e Busca em Profundidade (DFS)

9. ( ) A Busca em Profundidade pode ser usada para detectar ciclos em um grafo direcionado.
10. ( ) A Busca em Largura é utilizada no cálculo de componentes fortemente conexas em um grafo direcionado.
11. ( ) BFS sempre encontra o caminho mais curto em grafos não ponderados.
12. ( ) DFS é mais eficiente que BFS para encontrar o menor caminho em qualquer grafo.

Árvores Binárias

13. ( ) Uma árvore binária balanceada tem altura O(log n) no pior caso.
14. ( ) Em uma árvore binária de busca (BST), a remoção de um nó folha não requer realocação de nós.
15. ( ) O percurso **in-order** de uma árvore binária de busca sempre retorna os elementos em ordem decrescente.
16. ( ) A árvore AVL é um tipo de árvore binária balanceada, onde a diferença entre as alturas das subárvores de qualquer nó nunca pode ser superior a 2.

Algoritmos de Ordenação e suas Complexidades

17. ( ) O Quicksort tem tempo médio de execução O(n log n), mas no pior caso pode ser $O(n^2)$.
18. ( ) O algoritmo Heapsort tem complexidade O(n log n) no pior caso.

Equações de Recorrência

19. ( ) O método da substituição pode ser utilizado para resolver equações de recorrência.

Superposição de Qubits

20. ( ) Em um sistema quântico, um qubit pode representar simultaneamente 0 e 1 devido ao princípio da superposição.