1ª Questão

Acerto: 0,2 / 0,2

O método de ordenação da bolha, ou Bubblesort tem como melhor caso a entrada já ordenada, que resulta em complexidade O(n). Como seu pior caso, a entrada em ordem invertida, resultando em complexidade $O(n^2)$. Baseado nessas duas afirmações, podemos afirmar que a sua complexidade de caso médio é:

- ☐ O(nlog n)
- ☐ O(1)
- ☐ O(log n)
- ☐ O(n)
- ☒ ✔ $O(n^2)$

Respondido em 13/09/2023 16:02:34

Explicação:

Pelas características da notação O, a única afirmação que podemos extrair é que o caso médio é melhor ou igual ao pior caso. Portanto, é possível afirmar que o caso médio é $O(n^2)$, ou qualquer função assintoticamente superior a n^2, como $n^2 \log n$, n^3, 2^n etc.. Como dentre essas a única opção disponível é $O(n^2)$ essa é a resposta correta.

Podemos descartar O(1) e O(log n) por serem melhores que o melhor caso, o que contradiz a afirmativa do melhor caso.

Os casos O(n) e O(nlog n) seriam possíveis teoricamente para a complexidade média de um algoritmo qualquer que seja O(n) no melhor caso e $O(n^2)$ no pior caso, mas não é possível afirmar nenhuma das duas com as informações dadas.

De fato, o caso médio do Bubblesort é $O(n^2)$.

2ª Questão

Acerto: 0,2 / 0,2

Matrizes podem ser implementadas em Python utilizando a biblioteca numpy, trazendo diversas funções já implementadas. Dentre os pares de função com sua funcionalidade a seguir, qual é o correto?

- ☐ matriz.min() retorna o valor médio da matriz.
- ☐ matriz.max() retorna o desvio padrão da matriz.
- ☒ ✔ matriz.sum() retorna a soma dos elementos da matriz.
- ☐ matriz.std() retorna a variância da matriz.
- ☐ matriz.mean() retorna o valor mínimo da matriz.

Respondido em 13/09/2023 16:03:00

Explicação:

Dentre os pares apresentados, o único correto é o da função sum() que é a soma dos elementos. std() e mean() são funções estatísticas que retornam o desvio padrão e a média respectivamente. max() retorna o elemento de maior valor e min(), por sua vez, retorna o elemento de menor valor.

3ª Questão

Acerto: 0,2 / 0,2

No contexto de complexidade de algoritmos, usualmente é utilizada a notação O para representar as complexidades assintóticas analisadas. Dentre as afirmações a seguir, a correta é:

- ☐ O(n) significa que para $n=50$ o algoritmo realizará 50 operações no pior caso.
- ☐ O(n) significa que as operações variam em proporção logarítmica à entrada.
- ☒ ✔ $O(n^2)$ significa que as operações variam em proporção quadrática à entrada.
- ☐ O(n) significa que para $n=50$ o algoritmo executará no máximo 50 operações.
- ☐ c -O(log n) significa que para $n=64$ o algoritmo realizará 6 operações no pior caso.

Respondido em 13/09/2023 16:03:36

Explicação:

Com o uso da notação O, simplificamos o número de operações, ignorando multiplicadores constantes do termo dominante e todos os termos de menor complexidade. Por exemplo, $5n^2+3$ é $O(n^2)$, mas n^2 também é $O(n^2)$. Dessa forma, não é possível calcular exatamente o número de operações quando se usa a notação O. Apenas podemos fazer afirmações sobre a proporcionalidade ao tamanho da entrada n. Assim, a resposta correta é que $O(n^2)$ é proporcional ao quadrado da entrada.

4ª Questão

Acerto: 0,2 / 0,2

Uma Deque é uma estrutura de dados mais generalista que as pilhas e filas. Para implementá-la de forma eficiente, você pode usar:

- [ ] Fila com 2 variáveis: início e final.
- [x] Lista duplamente encadeada com 2 variáveis: início e final.
- [ ] Lista simplesmente encadeada com nó cabeça.
- [ ] Pilha com 1 variável: topo.
- [ ] Lista contígua com 1 variável: início.

Respondido em 13/09/2023 16:06:38

Explicação:

Para implementar uma deque eficientemente, você precisa ter um ponteiro para o início e o final da deque, permitindo inserções e remoções em ambas as pontas com complexidade O(1) , sem a necessidade de percorrer a estrutura, o que seria O(n).

Além disso, a fila é uma especialização da deque. Ou seja, toda fila é um deque, mas nem toda deque é uma fila. Podemos assim eliminar a resposta contendo fila. A resposta restante que possui 2 variáveis é a correta. Lista duplamente encadeada. Ela permite a inserção e remoção nas extremidades com complexidade O(1).

A lista contígua e a simplesmente encadeada com nó cabeça levariam a operação de inserção e remoção ao final da fila terem complexidade O(n) por precisarem percorrer toda a estrutura, sendo também descartadas.

5ª Questão

Acerto: 0,2 / 0,2

Em uma implementação da estrutura de dados do tipo fila, você possui um espaço de memória contíguo a ela alocada com capacidade para M nós. A variável da fila é F, e duas variáveis guardam os índices do início e final da fila (inicioF e finalF). Em uma implementação otimizada de F, como podemos identificar que a fila está cheia?

- [ ] InicioF = M
- [ ] InicioF== finalF
- [ ] FinalF== M
- [ ] InicioF==finalF + 1
- [x] InicioF==(finalF+1)mod M

Respondido em 13/09/2023 16:07:03

Explicação:

Em uma implementação otimizada da fila, é usado um sistema modular, onde o início e o final da fila se movem a cada inserção e remoção. A cada inserção, finalF aumenta em 1, até o máximo M, depois volta para 0 e assim por diante. A cada remoção inicioF aumenta em 1, até o máximo M e depois volta a 0. dessa forma a fila está cheia quando (finalF+1)modM é igual a início.

6ª Questão

Acerto: 0,0 / 0,2

Suponha que você está implementando um programa que precisa armazenar dados ordenados em uma estrutura para serem tratados posteriormente, na ordem inversa à que foram recebidos. Haverá uma grande quantidade de recebimentos e tratamento de dados, mas o tamanho esperado da estrutura não deve variar muito. Qual tipo de estrutura de dados é a melhor nessa situação?

- [ ] Lista duplamente encadeada.
- [ ] Pilha.
- [ ] Lista simplesmente encadeada.
- [ ] Lista em alocação contígua.
- [x] Fila.

Respondido em 13/09/2023 16:07:28

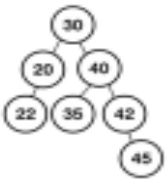
Explicação:

A pilha permite o tratamento de nós usando a política requerida, FILO ¿ ¿first in last out¿ -. Além disso, as operações de inserção e remoção são O(1), ou seja, de complexidade constante, a melhor possível. Isso condiz com o requisito de que haverá muitas operações desse tipo. Por fim, o fato de a estrutura não variar muito em tamanho permite o uso de uma alocação contígua e otimizada para a pilha. A fila não obedece a lógica FILO e as listas têm complexidade de inserção e remoção O(n) sendo muito piores que a pilha, principalmente quando o número desses tipos de operação é grande.

7ª Questão

Acerto: 0,2 /0,2

Seja a seguinte árvore, marque a opção correta que indica o porquê a árvore abaixo não é uma árvore binária de busca:



- [x] Não é árvore binária de busca pois o nó 22 deveria estar inserido à direita do nó 20.
- [ ] Não é árvore binária de busca pois está desbalanceada.
- [ ] Não é árvore binária de busca pois essa árvore deve estar perfeitamente balanceada.
- [ ] Não é árvore binária de busca pois o nó 35 deveria estar inserido à direita do nó 20.
- [ ] Não é árvore binária de busca pois esta árvore deve estar com os níveis de suas folhas todas igualmente perfeitas.

Respondido em 13/09/2023 14:07:31

Explicação:

Uma árvore binária de busca são árvores que obedecem às seguintes propriedades:

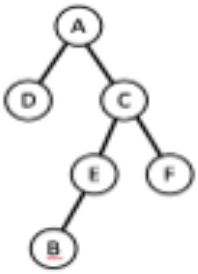
- Dado um nó qualquer da árvore binária, todos os nós à esquerda dele são menores ou iguais a ele.
- Dado um nó qualquer da árvore binária, todos os nós à direita dele são maiores ou iguais a ele.

Observe que a sub-árvore 20-22 não respeita a regra básica, portanto, o nó 22 deveria estar a direita do nó 20.

8ª Questão

Acerto: 0,2 /0,2

Seja a seguinte árvore binária de busca, marque a opção que apresenta o percurso em pré-ordem dessa árvore:



- [x] A,D,C,E,B,F
- [ ] D,B,E,F,C,A
- [ ] D,A,B,E,C,F
- [ ] F,C,E,B,A,D
- [ ] A,B,C,D,E,F

Respondido em 13/09/2023 14:08:21

Explicação:

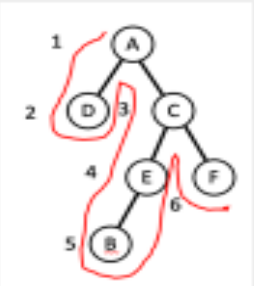
A resposta correta é a questão B. O percurso em pré-ordem é definido como se segue. A partir da raiz *r* da árvore *T*, percorre-se a árvore da seguinte forma:

1 - visita-se a raiz;

2 - percorre-se a subárvores esquerda de *T* em pré-ordem e

3 - percorre-se a subárvores direita de *T* em pré-ordem.

Resultado da pesquisa pré-ordem:



9ª Questão

Acerto: 0,2 / 0,2

As afirmativas abaixo são feitas com base na estrutura de dados "Árvore Binária de Busca". Em relação ao algoritmo de busca em uma árvore binária de busca, analise as afirmativas abaixo:

I -A complexidade da busca é definida pela altura da árvore binária de busca. No pior caso O(n).

II - A busca é definida de forma recursiva, parte da raiz, comparando a chave buscada com a armazenada na raiz, caso seja igual temos o sucesso da busca, caso contrário, se a chave buscada for menor, devemos proceder recursivamente no ramo esquerdo, se a chave buscada for maior, proceder recursivamente no ramo direito.

III - Sempre é necessário percorrer toda a árvore no algoritmo de busca. Em todos os casos, mesmo em árvores completas.

IV - A condição de parada da busca é encontrar a chave buscada ou ter que descer por um ramo vazio.

V - É possível escrever o algoritmo da busca de forma não recursiva.

- ☐ I, II, III, IV e V são corretas.
- ☐ II, III, IV e V são corretas.
- ☐ I, III, IV e V são corretas.
- ☐ I, II, III e IV são corretas.
- ☒ ✔ I, II, IV e V são corretas.

Respondido em 13/09/2023 16:09:09

Explicação:

A afirmativa III é incorreta, não é necessário percorrer todos os nós da árvore se ela estiver perfeitamente balanceada.

10ª Questão

Acerto: 0,2 / 0,2

As árvores AVL constituem uma importante estrutura de dados que disponibilizam operações de busca, inserção e remoção. Classifique como verdadeiro ou falso as afirmativas abaixo:

I - As árvores de Fibonacci são as árvores de altura máxima h com número mínimo do nós n e altura proporcional a log n.

II - As árvores completas são árvores AVL.

III - É possível construir uma topologia de uma árvore AVL que não seja nem completa nem de Fibonacci com altura proporcional a log n.

IV - Uma vez que a altura das árvores AVL é proporcional a log n, podemos garantir que a busca ocorre numa complexidade de O(log n).

V - Na remoção, pode ser necessário realizar todas as rotações, no pior caso, do pai de uma folha que está sendo removida até a raiz. Por esta razão, a complexidade da remoção é maior que O(log n).

- ☐ I-V, II-F, III-F, IV-V, V-V.
- ☐ I-F, II-F, III-F, IV-V, V-V.
- ☐ I-F, II-F, III-V, IV-F, V-F.
- ☐ I-V, II-V, III-F, IV-V, V-F.
- ☒ ✔ I-V, II-V, III-V, IV-V, V-F.

Respondido em 13/09/2023 16:12:10

Explicação:

Nem sempre é necessário realizar todas as operações, visto que a remoção pode eliminar uma folha e não causar desbalanceamento na árvore.