

Revisar envio do teste: Semana 4 - Atividade Avaliativa

Usuário	LIZIS BIANCA DA SILVA SANTOS
Curso	Algoritmos e Programação de Computadores II - COM120 - Turma 003
Teste	Semana 4 - Atividade Avaliativa
Iniciado	18/05/23 12:58
Enviado	18/05/23 13:04
Data de vencimento	19/05/23 05:00
Status	Completada
Resultado da tentativa	10 em 10 pontos
Tempo decorrido	5 minutos
Instruções	Olá, alunos e alunas!
Esta atividade possui múltipla escolha. Para respondê-la:	
1. Selecione, com o mouse, a alternativa que você considerar correta;	
2. Repare que, ao selecionar uma alternativa, as seleções anteriores são desmarcadas;	
3. Após selecionar a resposta correta em todas as questões, vá até o fim da página e clique em "Enviar teste".	
Pronto! Sua atividade já está registrada no AVA.	
Resultados exibidos	Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,44 em 1,44 pontos



As listas são estruturas organizadas em uma sequência de elementos que pertencem ao mesmo tipo (como *string*, *int* e *float*) e têm o mesmo tamanho, sendo armazenadas em posições contíguas de memória. Em Python, essa é a estrutura de dados mais básica e armazena elementos em sequência, em que cada um contém um índice (sua posição na lista) que é sempre 0 para o primeiro elemento, sendo incrementado a cada elemento inserido na lista. A escolha entre as distintas implementações possibilita melhorar a aplicação e o espaço de alocação de memória. Considerando as distintas formas para implementação de listas, avalie as afirmações a seguir e relacione-as adequadamente aos termos ou ferramentas às quais se referem.

1. Alocação sequencial.
2. Alocação encadeada.
3. Alocação estática.
4. Alocação dinâmica.

- I. Os dados da lista não exigem um tamanho fixo devido à possibilidade de definir a quantidade de memória que se pretende utilizar.
II. Os dados da lista estão dispostos de forma sequencial na memória e possuem um tamanho fixo.
III. Permitem a realização de um acesso aleatório a uma determinada posição da lista, pois os índices são usados para o acesso direto a uma posição de memória.
IV. Disponibiliza os elementos que não se encontram necessariamente em posições adjacentes de memória, mas considera uma posição lógica.

Assinale a alternativa que correlaciona adequadamente os dois grupos de informação.

Resposta Selecionada: ☒ b. 1-III; 2-IV; 3-II; 4-I.

- Respostas:
- a. 1-IV; 2-III; 3-II; 4-I.
- ☒ b. 1-III; 2-IV; 3-II; 4-I.
- c. 1-I; 2-II; 3-III; 4-IV.
- d. 1-I; 2-III; 3-II; 4-IV.
- e. 1-II; 2-I; 3-VI; 4-III.

Comentário da resposta:

JUSTIFICATIVA

A sentença I se enquadra no conceito 4, pois é um dos grupos principais para alocação de memória no computador, em que os elementos não necessitam de um tamanho fixo já que pode ser definida quantidade de memória para cada elemento, portanto pode-se alocar espaços de memória que não precisam estar organizados na forma sequencial e podem estar distribuídos dispersamente na memória. É solicitado que os espaços sejam alocados ou desalocados acordando com a necessidade.

A sentença II se enquadra no conceito 3, visto que é um dos grupos principais para alocação de memória no computador, a alocação estática possui os elementos com um tamanho fixo e organizados sequencialmente na memória do computador, as variáveis globais podem ser citadas como um exemplo de alocação estática.

A sentença III se enquadra no conceito 1, já que, em Python, as listas comuns declaradas ao abrir e fechar colchetes são listas lineares em alocação sequencial e permite o acesso aleatório a uma posição específica da lista com o uso de índices (p. ex.: lista[3]) que acessa diretamente uma posição de memória que contém o quarto elemento da lista, pois se começa a partir do 0 em Python. A ordem lógica dos elementos (ordem visível ao usuário) é a mesma ordem em memória principal, o que indica que elementos vizinhos na lista se encontram em posições vizinhas de memória.

A sentença IV se enquadra no conceito 2, porque a alocação encadeada em relação à quantidade de elementos é flexível, podendo ser adaptada para modelar distintas necessidades sendo bastante utilizada para armazenar de forma temporária conjuntos de elementos, aqui os elementos não estão necessariamente em posições adjacentes de memória.

Pergunta 2

1,44 em 1,44 pontos



As estruturas de dados organizadas em pilhas permitem inserir e remover um elemento do final da pilha construída, o que indica que o último elemento que foi inserido na estrutura será o primeiro elemento a ser removido da estrutura, e para que isso aconteça normalmente a implementação é feita com arranjos utilizando métodos para inserção e para remoção.

Assinale a alternativa que representa o método destinado à inserção de elementos na pilha.

Resposta Selecionada: ☒ b. append()

- Respostas:
- a. empty()
- ☒ b. append()
- c. popleft()
- d. top()
- e. pop()

Comentário da resposta:

JUSTIFICATIVA

Em Python, a lista de estrutura de dados embutida pode ser usada como uma pilha, portanto ao invés do comando push() que adiciona um elemento no topo de uma pilha, faz-se uso do append() que é padrão de listas em Python para adicionar elementos ao final da pilha. E para remover o elemento na ordem LIFO, utiliza-se pop(). Já o popleft() retorna o elemento removido em uma fila, enquanto empty() indica se a variável é vazia, retornando true caso seja e false caso a variável não seja vazia. E para saber o elemento mais alto da pilha, usa-se top() que retorna uma referência a esse elemento.

Pergunta 3

1,42 em 1,42 pontos



Uma árvore binária (T) é um conjunto finito de n nós que pode ser nula (T é um conjunto vazio o que indica n=0) ou ter características como: a existência de um nó especial chamado raiz; nós restantes divididos em T1, T2 estruturas disjuntas em que T1 é a subárvore esquerda e T2 é a subárvore direita da raiz. Sua utilização pode ser a avaliação de expressões.

Avalie as alternativas abaixo e indique a que representa o conteúdo de cada nível de uma árvore para a expressão (6 + 9) * (3 - 7) + 8.

Resposta Selecionada: ☒ d. Primeiro nível: +
Segundo nível: * 8
Terceiro nível: + -
Quarto nível: 6 9 3 7

- Respostas:
- Primeiro nível: 6 9 3 7
Segundo nível: 8
Terceiro nível: *
a. Quarto nível: + -
- Primeiro nível: *
Segundo nível: +
Terceiro nível: 8 -
b. Quarto nível: 6 9 3 7
- Primeiro nível: 6 9
Segundo nível: 8
Terceiro nível: 3 7
c. Quarto nível: + * -
- Primeiro nível: +
Segundo nível: * 8
Terceiro nível: + -
☒ d. Quarto nível: 6 9 3 7
- Primeiro nível: 6 9
Segundo nível: 3 7
Terceiro nível: 8
e. Quarto nível: + * -

Comentário da resposta:

JUSTIFICATIVA

Como exemplo do uso de árvores binárias, pode ser citada a avaliação de expressões. Na árvore que representa a expressão (6 + 9) * (3 - 7) + 8, os nós-folhas representam operandos e os nós internos, operadores binários. Dessa forma, os nós são apresentados em níveis que começam de cima e vão abaixo, e o primeiro nível contém o nó com +. O segundo nível contém nós com * e 7 cada. O terceiro nível contém nós + e - cada. O quarto nível contém nós com 5, 9, 3 e 7 cada.

Pergunta 4

1,42 em 1,42 pontos



A implementação de uma fila ou de uma pilha deve oferecer um bom desempenho por compor programas usados diariamente por usuários em seus cotidianos, assim, da mesma forma para as filas, a implementação de pilhas pode utilizar uma lista como estrutura para armazenamento dos dados, exigindo que se defina como será o funcionamento de tal estrutura.

Após análise da situação apresentada, avalie as asserções a seguir e a relação proposta entre elas.

I - Para que uma fila seja manipulada corretamente, deve-se considerar métodos adicionais e restrições devido aos vários tipos de fila como Queue (FIFO, *First In, First Out*), LifoQueue (*Last In, First Out*) e PriorityQueue (por ordem crescente de valor).

POIS

II - O módulo integrado do Python queue pode ser utilizado tanto para filas quanto para pilhas de forma a oferecer um apoio na programação encadeada quando as informações devem ser trocadas com segurança entre vários encadeamentos.

A respeito dessas asserções, assinale a alternativa correta.

Resposta Selecionada: ☒ b. As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I

- Respostas:
- a. As asserções I e II são falsas.
- ☒ b. As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I
- c. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- d. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- e. As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.

Comentário da resposta:

JUSTIFICATIVA

A asserção I é verdadeira, o módulo integrado do Python é útil na programação encadeada porque auxilia no processo em que as informações devem ser trocadas de forma segura entre os encadeamentos, implementando 3 tipos de filas cuja diferença entre elas se concentra na ordem em que as entradas são recuperadas.

A asserção II é verdadeira, no caso da FIFOQueue, os primeiros elementos inseridos são os primeiros a serem removidos. Na LIFOQueue, o elemento que foi adicionado por último é o primeiro a ser removido (operando como uma pilha). A asserção II justifica a primeira, e no caso de uma fila de prioridade, os elementos inseridos são mantidos ordenados e o elemento de menor valor é o primeiro a ser removido.

Pergunta 5

1,42 em 1,42 pontos



Uma árvore apresenta uma estrutura hierárquica de vários elementos, ou seja, composta por um conjunto de _____ e servem, por exemplo, para organizar informações em sistemas de banco de dados, pois permitem um tratamento computacional eficiente sendo mais fácil de manipular. Há um nó chamado _____ que possuem zero ou mais subárvores onde suas raízes estão ligadas a r, e os nós-raízes dessas subárvores são chamados _____ do nó _____ r e as folhas são nós com grau zero.

Preencha as lacunas escolhendo a alternativa correta.

Resposta Selecionada: ☒ e. nós, raiz, filhos, pai.

- Respostas:
- a. filhos, pai, nós, raiz.
- b. nós, filhos, pai, raiz.
- c. filhos, raiz, pai, nós.
- d. raiz, pai, filhos, nós.
- ☒ e. nós, raiz, filhos, pai.

Comentário da resposta:

JUSTIFICATIVA

Na primeira lacuna, é correto preencher com o termo "nós", pois é onde os dados ficam armazenados.

Na segunda lacuna, é correto preencher com o termo "raiz", porque é o nó que ocupa o primeiro lugar na estrutura hierárquica da árvore, situando-se no topo.

Na terceira lacuna, é correto preencher com o termo "filhos", pois esses elementos ficam um nível abaixo do nó-pai formando uma subárvore.

Na quarta lacuna, é correto preencher com o termo "pai", que é o elemento que possui os nós-filhos, caso contrário (se não possuírem filhos) são chamados de "folhas".

Pergunta 6

1,43 em 1,43 pontos



O uso das listas é recomendado para aplicações em que não há como prever a quantidade de memória a ser usada, possibilitando manipular quantidades inesperadas de dados. Considere o trecho da estrutura abaixo, em que: x' com tamanho suficiente para armazenar a estrutura foi usado para depositar os elementos; o campo ULTIMO vai apontar para a próxima posição após o último elemento; MaxTam determina o máximo tamanho que a lista pode ter.

```
1 #define INICIO
2 #define MAXTAM 1000
3 typedef int TipoApontador;
4 typedef int TipoChave;
5 typedef struct
6 {
7     TipoChave Chave;
8     /* --- outros componentes--- */
9 }
10 } TipoElemento;
11 typedef struct
12 {
13     TipoElemento Elemento[MAXTAM] ;
14     TipoApontador Primeiro , Ultimo;
15 } TipoLista;
```

Fonte: Elaborada pela autora.

Assinale a alternativa que indica a forma de implementação citada.

Resposta Selecionada: ☒ c. Implementação de listas por meio de arranjos.

- Respostas:
- a. Implementação de pilhas por meio de arranjos.
- b. Implementação de filas por meio de apontadores.
- ☒ c. Implementação de listas por meio de arranjos.
- d. Implementação de listas por meio de arranjos.
- e. Implementação de listas por meio de apontadores.

Comentário da resposta:

JUSTIFICATIVA

Na implementação de listas por meio de arranjos, os dados são posicionados na memória em posições contíguas possibilitando que a varredura aconteça em qualquer sentido, um novo item, por exemplo, é inserido após o último elemento e caso a inserção ocorra no meio da lista é necessário o deslocamento dos itens após o ponto de inserção, igualmente acontece ao remover o item do início da lista que também requer um deslocamento de itens para preencher o espaço que fica vazio.

A forma de implementação de listas por meio de apontadores não se enquadra na forma de implementação citada, pois os dados são armazenados em posições não contíguas de memória com a inserção e remoção de itens sem exigir deslocar os itens seguintes da lista.

A forma de implementação de filas por meio de apontadores não se enquadra na forma de implementação citada, pois uma célula cabeça facilita a implementação das operações Enfileira e Desenfileira quando a fila está vazia, e os apontadores Frente e Trás apontam para a célula cabeça.

A forma de implementação de pilhas por meio de arranjos não se enquadra na forma de implementação citada, pois os itens são armazenados em posições contíguas de memória, quando a operação Enfileira expande a parte de trás da fila e a operação Desenfileira contrai a parte da frente da fila.

A forma de implementação de pilhas por meio de arranjos não se enquadra na forma de implementação citada, porque os itens são armazenados em posições contíguas de memória e a implementação para inserção e remoção de itens acontecem no topo da pilha.

Pergunta 7

1,43 em 1,43 pontos



Nas estruturas não sequenciais os elementos não são individualizados por meio de seu índice. Portanto utilizam meios que possibilitam o acesso a esses elementos, tais meios são coleções de dados que permitem, por exemplo, adicionar, remover, acessar, pesquisar e ordenar. _____ são coleções de dados não ordenados que comporta elementos _____ e imutáveis (como *string*, *int* e tuplas) e possuem métodos para operações matemáticas (união, interseção, diferença e diferença assimétrica). Os _____ são não sequenciais e compreendem uma chave e um valor, sendo que cada chave possui um valor associado, o que possibilita o acesso à um elemento individual, portanto, devem ser utilizados diante da necessidade de armazenamento de dados de maneira organizada.

Preencha as lacunas escolhendo a alternativa correta.

Resposta Selecionada: ☒ a. sets, únicos, dicionários.

- Respostas:
- ☒ a. sets, únicos, dicionários.
- b. sets, múltiplos, dicionários.
- c. dicionários, múltiplos, sets.
- d. sets, mutáveis, dicionários.
- e. dicionários, únicos, sets.

Comentário da resposta:

JUSTIFICATIVA

Na primeira lacuna, o correto é preencher com "*sets*", pois abrangem coleções de dados não ordenados que comporta elementos únicos e imutáveis e possuem métodos para operações matemáticas (união, interseção, diferença e diferença assimétrica), cabe ressaltar que *sets* são mutáveis, por isso permitem alterar, excluir ou adicionar itens e são recomendados no trabalho com elementos únicos ou diante da necessidade de se realizar operações matemáticas com elementos de diferentes *sets*.

Na segunda lacuna, o correto é preencher com "únicos", porque os *sets* são coleções de dados não ordenados que comportam elementos únicos e imutáveis.

Na terceira lacuna, o correto é preencher com "dicionários", visto que trata-se de estruturas que compreendem um conjunto de pares (chave e valor) e cada chave individual possui um valor associado (isso leva o nome de "coleção associativa desordenada") sendo estruturas mutáveis. Como são não sequenciais, o acesso a um determinado elemento ocorre por meio de uma chave.