**Prova 2**

**Estrutura de Dados**

Olá pessoal, como vão?

Sim, a prova está trabalhosa e não difícil. É fazível, porém precisará suar um pouco.

A prova vale 11, sim 11 pontos e não 10. E ainda tem algumas questões extras lá no final para vocês fazerem e adicionarem mais pontos na prova.

Acertando toda a prova(que é possível sim) poderá tirar 15 pontos.

Entender a questão é tão importante quanto as respostas dadas.

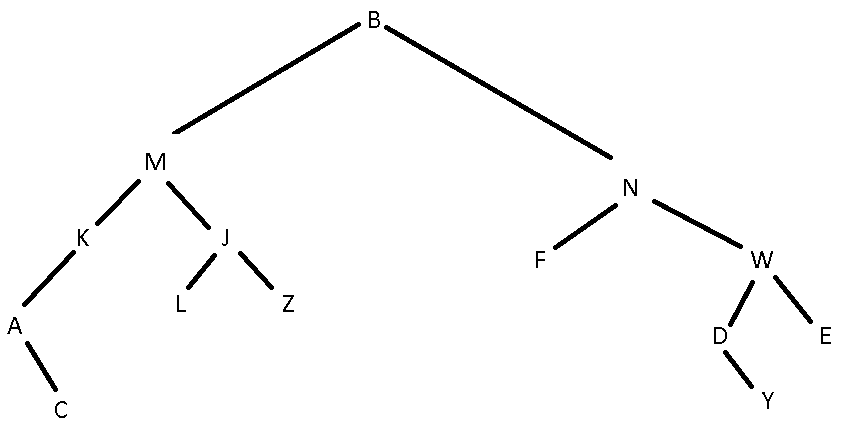
Nas questões que precisarem desenhar: desenha no papel (ou paint) -> transforma em imagem e anexa aqui no word (na questão correspondente é claro).

Subir arquivo word com o seu nome no nome do arquivo.

**Bons estudos e boa prova!!**

**Questão 1) 0.5 ponto**

Desenhe uma única árvore binária que apresenta os percursos descritos abaixo.  
Pré-ordem: B M K A C J L Z N F W D Y E  
Em-ordem: A C K M L J Z B F N D Y W E



**Questão 2) 1.0 ponto**

Montar uma ABB, inicialmente vazia, com as letras do **seu** nome. **Por exemplo**, usando o nome do professor, deveria inserir os seguintes elementos (nessa ordem): S, A, M, A, R, O, U, H, A, N, I.

Então, inserir e desenhar as letras do seu nome em uma Árvore Binária. Após desenhar, responda as questões abaixo:

* Qual nó tem o maior grau e qual o grau da árvore.

O 1º nó, ele possui grau 2.

* Qual a altura/profundidade da árvore

nível 7.

* É possível calcular a altura da sua árvore? Sim/Não e porque?

Sim, pois os níveis estão bem evidentes.

* É possível calcular a quantidade de “nós” da sua árvore? Sim/Não e porque?

Sim, mas sempre será até dois por ser binária.

* É uma árvore balanceada? Sim/Não e porque?

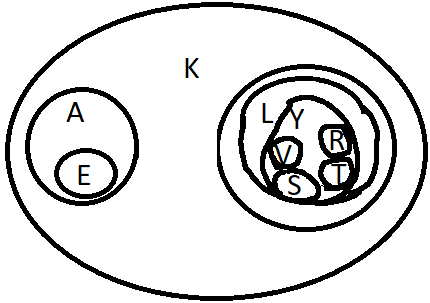
Não, porque há uma diferença maior que apenas um nó.

* Qual a sequência de nós no percurso Pré-ordem, Em-ordem e Pós-ordem (use o macete que passei em sala).Represente a árvore com parênteses.

((A(E))(K(L(Y(R)(S)(T)(V)))))

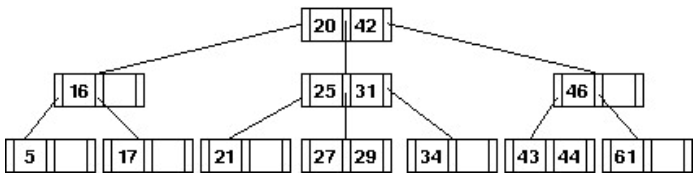
* Represente a árvore com os Diagramas de Venn

(Klayvert Alves)

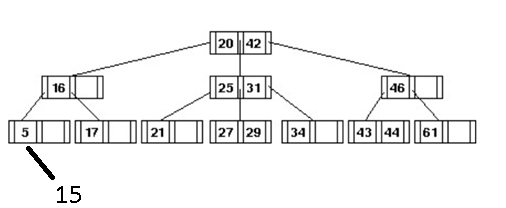


**Questão 3) 0.5 ponto**

Indique na árvore abaixo uma chave cuja inserção provocará o aumento da altura total da árvore. Desenhe a árvore resultante após a inserção.



Inserir o número 15.



**Questão 4) 1.0 ponto**

Montar uma ABB, inicialmente vazia, inserindo um por um de cada número do **seu** RA, porém irá repetir 3 vezes o seu RA, **nessa ordem**: 10\*RA, 5\*RA, 1\*RA

**Por exemplo**, usando um RA fictício 2 9 0 1 3 9 2 2 1 3 0 2 1, será assim:

Meu RA: 2901392213018

10\*RA: 20 90 0 10 30 80

5\*RA: 10 45 0 5 15 40

1\*RA: 290138

Observe que o primeiro conjunto de números é **cada** número do RA multiplicado por 10, o segundo conjunto é **cada** número do RA multiplicado por 5 e o terceiro conjunto é somente os números do seu RA.

Os números que repetirem não serão inseridos.

Então, inserir e desenhar as letras do seu nome em uma Árvore Binária. Após desenhar, responda as questões abaixo:

* Qual nó tem o maior grau e qual o grau da árvore.

1 nó, grau 2.

* Qual a altura/profundidade da árvore

Nível 6.

* É possível calcular a altura da sua árvore? Sim/Não e porque?

Pode porque ela possui diversos níveis.

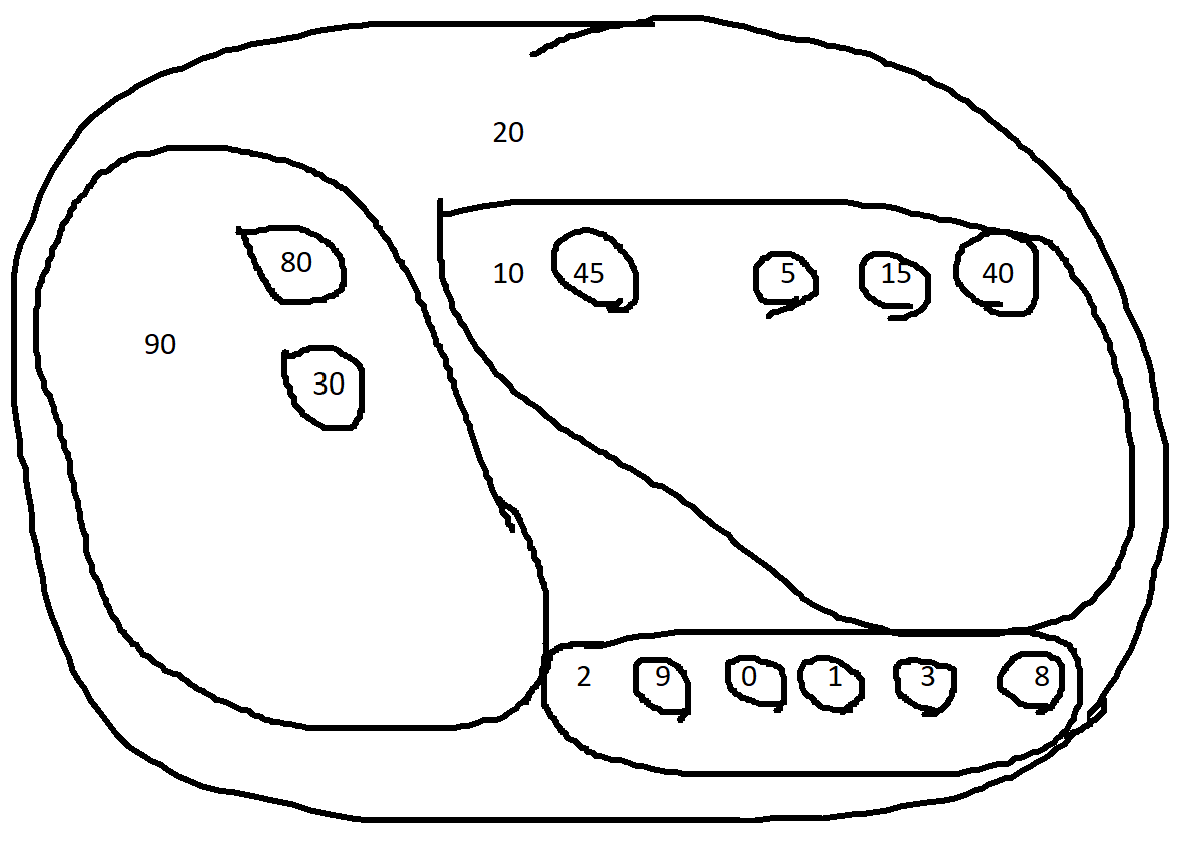
* É possível calcular a quantidade de “nós” da sua árvore? Sim/Não e porque?

É possível pelo simples fato de ela ser binária, logo, não terá mais que dois nós.

* É uma árvore balanceada? Sim/Não e porque?

Não porque ela tem diferença de mais de um nó.

* Qual a sequência de nós no percurso Pré-ordem, Em-ordem e Pós-ordem (use o macete que passei em sala). Represente a árvore com parênteses.
* ((0)(1)(2(3))(5)(8)(9)(10(15)(20(30)(40)(45)(80)(90))))Represente a árvore com o Diagrama de Venn



**Questão 5) 0.5 ponto**

O caminho com percurso pós-ordem em uma árvore binária resultou na sequência “A X K D C J B”, em que cada caractere refere-se a um nó visitado. Nesse caso, o nó raiz refere-se ao caractere.

1) A

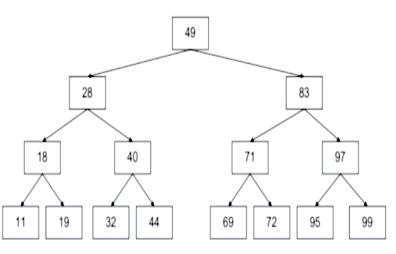
2) C

1. D
2. B
3. K

Alternativa (D) caractere B.

**Questão 6) 0.5 ponto**

Indique o percurso dos nós em ordem simétrica (em ordem) da árvore binária abaixo:



* 49 - 28 - 18 - 11 - 19 - 40 - 32 - 44 - 83 - 71 - 69 - 72 - 97 - 95 - 99x
* 11 - 18 - 19 - 32 - 28 - 40 - 44 - 49 - 69 - 72 - 71 - 83 - 95 - 97 - 99x
* 11 - 18 - 19 - 28 - 32 - 40 - 44 - 49 - 69 - 72 - 71 - 83 - 95 - 97 - 99x
* 11 - 19 - 18 - 32 - 44 - 40 - 28 - 69 - 72 - 71 - 95 - 99 - 97 - 83 - 49x
* 11 - 18 - 19 - 28 - 32 - 40 - 44 - 49 - 69 - 71 - 72 - 83 - 95 - 97 - 99v

alternativa (e)

* **Questão 7) 1.0 ponto**
* O que é uma ABC (Árvore Binária Completa)?

Nós com menos de 2 filhos estão no final da árvore e os nós no último nível são todos alinhados à esquerda

* Se temos uma ABC,de altura 10, ou seja, h=10, então qual será a quantidade de nós dessa árvore?

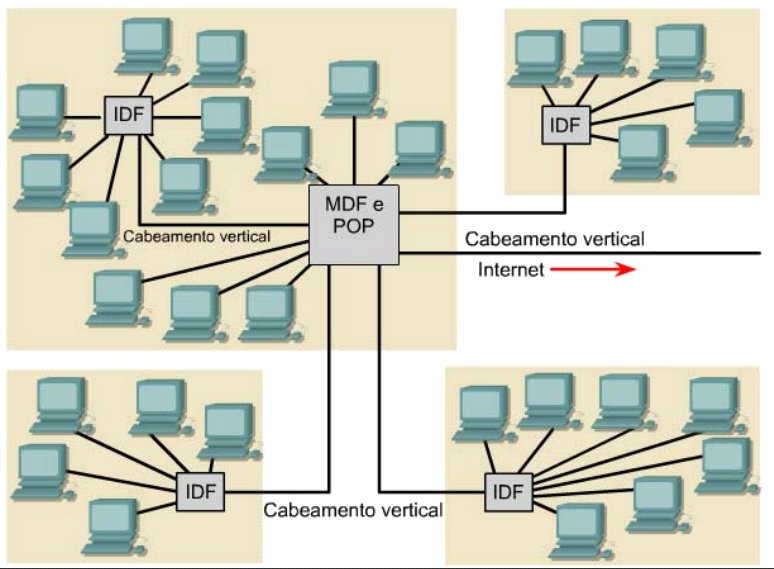
2(^10)-1 = 1023.

* E se temos uma ABC com 31 “nós”, então qual será a altura h da nossa árvore?

Altura = 5.

**Questão 8) 1.0 ponto--------------------------------------------------------------------------**

Analisando a imagem abaixo de cabeamento de rede de computadores, crie o seu correspondente grafo. Depois de criar o grafo, responda as questões abaixo:



Obs 1: desconsidere a aresta escrita “Cabeamento vertical internet”

Obs 2: dê uma numeração aos vértices para te ajudar nas questões.

* Qual a quantidade de vértices e arestas do grafo?

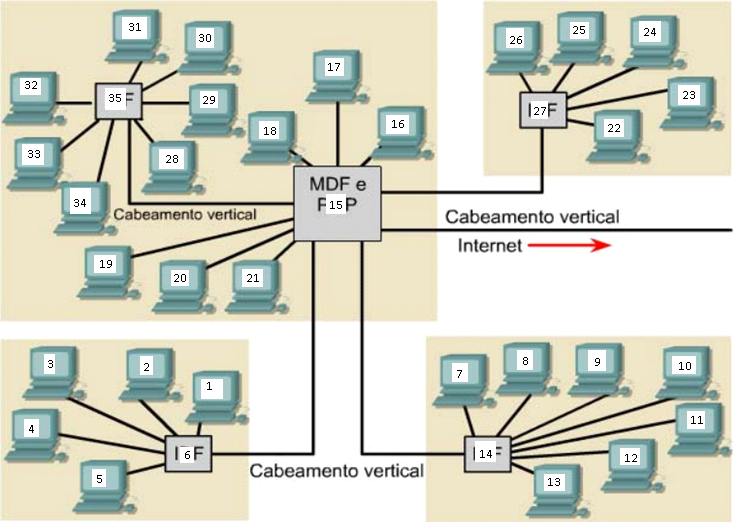
v(G) = 34

a(G) = 33

* Qual a ordem do grafo?

ordem(G) = 34

* Escreva a representação V(G) e E(G) do grafo



V(G): V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, V25, V26, V27, V28, V29, V30, V31, V32, V33, V34. V35

E(G):

(V6,V1); (V6,V2); (V6,V3); (V6,V4); (V6,V5);

(V14.V7); (V14,V8); (V14,V9); (V14,V10); (V14,V11); (V14,V12); (V14,V13);

(V14,V15); (V6,V15);

(v15,V16); (v15,17); (v15,18); (v15,V19); (v15,V20); (v15,V21);

(v35,V28); (v35,V29); (v35,V30); (v35,V31); (v35,V32); (v35,V33); (v35,V34);

Monte a matriz de adjacência do grafo

* Monte a lista de adjacência do grafo

**Questão 9) 1.0 ponto--------------------------------------------------------------------------**

Analisando o dígrafo abaixo, responda as questões que se seguem:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | * É um grafo regular ou irregular?   Ele é irregular   * Escreva a representação V(G) e E(G) do   dígrafo  V(G) = 1,2,3,4,5,6,7,8,9,10  E(G) = (1,2);(1,4);(1,6);(1,7);  (2,3);(2,4);(3,2);(3,4);(4,5); (5,6);  (7;6);(8,1);(8,7);(9,2);(9,8);(9,10);  (10,2);(10,3);     * Monte a matriz de adjacência do dígrafo,   com os respectivos pesos   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | \* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 1 | 0 | 10 | 0 | 20 | 20 | 5 | 15 | 0 | 0 | 0 | | 2 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | | 3 | 0 | 15 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | | 4 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | | 5 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 7 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | | 8 | 5 | 20 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | | 9 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 10 | | 10 | 0 | 5 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  * Monte a lista de adjacência do dígrafo,   com os respectivos pesos.  1 5 2,4,5,6,7  2 2 3,4  3 2 2,4  4 1 5  5 1 6  6 0 0  7 1 6  8 3 1,2,7  9 3 2,8,10  10 2 2,3   * Represente o din e dout para cada vértice.   din(V1) =1 dout(V1) =5,  din(V2) =5 dout(V2) =2,  din(V3) =2 dout(V3) =2,  din(V4) =4 dout(V4) =1,  din(V5) =2 dout(V5) =1,  din(V6) =3 dout(V6) =0,  din(V7) =2 dout(V7) =1,  din(V8) =1 dout(V8) =3,  din(V9) =0 dout(V9) =3,  din(V10) =1 dout(V10) =2   * Quem são os vértices fonte e sumidouro   do dígrafo  fonte = 9  sumidouro = 6 |

**Questão 10) 0.5 ponto**

Com relação ao grafo abaixo responda as questões.

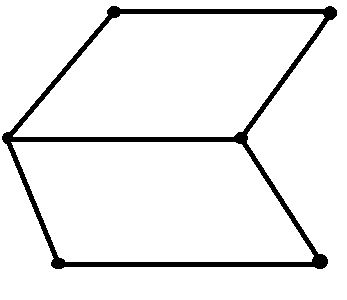
|  |  |
| --- | --- |
|  | * Qual a ordem do grafo. * 11 * É um grafo simples ou não e porque. * Ele tem mais de 1 aresta ligando 2 vértices, logo, não é simples. * Se fosse um grafo completo teria quantas arestas. * 55 * É um grafo regular ou irregular e porque. * Irregular pois as arestas não possuem o número exato de ligações * Qual seria o caminho Hamiltoniano e Euleriano? * Hamiltoniano: passa por todos os vértices  apenas uma vez * Faça o complemento desse grafo |

**Questão 11) 0.5 ponto**

Quantas arestas tem um grafo com vértices de graus 5, 2, 2, 2, 2, 1? Desenhe um possível grafo.

5+2+2+2+2+1=14

v=6; a=7;



**Questão 12) 0.5 ponto**

Em relação ao código de remoção de nós da ABB dado em sala, explique a grosso modo as 3 partes do código e as 2 estratégias de remoção existentes.

Caso1:

O nó(a) a ser removido não tem filhos.

O nó(a) é removido.

O nó(a) aponta para NULL.

Caso2(nó com filho único):

Move o filho para o lugar do pai

Remove o nó

Caso3(nó com 2 filhos):

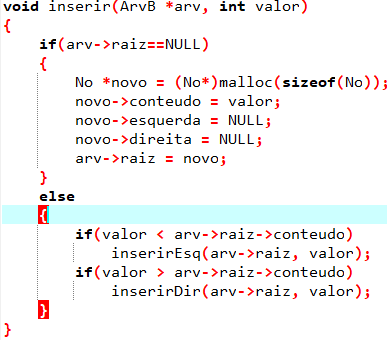
Localiza a menor sub-árvore da direita ou maior da esqueda

O nó pai “pega” esse valor

O nó que estava ocupado recebe free

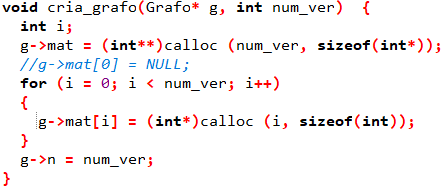
**Questão 13) 0.5 ponto-------------------------------------------------------------------------**

Explique o que faz o código abaixo e onde ele se insere no contexto da matéria.



**Questão 14) 0.5 ponto**

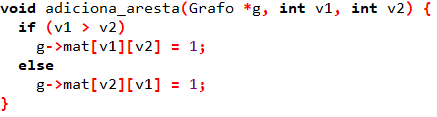
Explique o que faz o código abaixo e onde ele se insere no contexto da matéria.



Esse código é usado para criar o gráfico, alocar espaço para o gráfico e, em seguida, alocar espaço para os vértices. Então o número de vértices que o nó if começa a apontar

**Questão 15) 0.5 ponto**

Explique o que faz o código abaixo e onde ele se insere no contexto da matéria.



Este código é usado para adicionar uma aresta representada pelo número 1 à figura de um meio quadrado (triângulo), e toda vez que as coordenadas marcam o outro lado do quadrado (outro triângulo), ele inverte as coordenadas e o envia para o "direita" "Um lado

**Questão 16) 1 ponto**

Diferencie um grafo Hamiltoniano de um Euleriano. Explique cada um deles.

Um grafo conexo G é Euleriano se e somente se todo vértice de G tem um grau par. Ou seja, se o grafo é um grafo de Euleriano, então o grau de todos os vértices é par, além disso, se o grau de todos os vértices do grafo é par, então o grafo é um grafo de Euleriano.

Um ciclo hamiltoniano na teoria dos grafos é um caminho em um grafo não direcionado que visita cada vértice apenas uma vez. Um ciclo hamiltoniano (ou loop) visita cada vértice do grafo apenas uma vez, terminando no vértice inicial. Determinar se tal caminho (ou ciclo) existe em um grafo é um problema NP-completo.

**QUESTÕES EXTRAs**

**Questão extra 1) 1.0 ponto**

Faça as associações corretas para as estruturas de dados abaixo

I – O primeiro a entrar é o último a sair

II – O primeiro a entrar é o primeiro a sair

III - “é uma sequência de células, sendo que cada célula contém um objeto e o endereço da célula seguinte; todos os objetos são do mesmo tipo”.

IV - permite a inserção e a eliminação de elementos em ambas as extremidades

V - passagem de parâmetro a um procedimento em que tal procedimento recebe a cópia dos parâmetros passados.

VI - conjunto finito de nós e arestas interligando os nós.

A) – FIFO

B) – FILO

C) – grafo

D) – passagem de parâmetro por valor

E) – lista encadeada

F) - deque

* I – A, II – B, III – C, IV – D, V – E, VI - F
* I – B, II – C, III – D, IV – A, V – F, VI - E
* I – B, II – A, III – E, IV – F, V – D, VI – C
* I – C, II – D, III – F, IV – E, V – A, VI – B
* I – E, II – F, III – C, IV – A, V – B, VI - D

Alternativa c)

**Questão extra 2) 1.5 ponto**

Em relação aos métodos de ordenação, descreva sucintamente cada um deles (3 linhas por método), falando de suas vantagens e desvantagens e como funciona cada método. São 6 métodos para descrever (bubble sort, insertion sort, selection sort, shell sort, quick sort e merge sort).

Bubble sort é um algoritmo de ordenação comumente usado em ciência da computação. O Bubble Sort é baseado na ideia de comparar repetidamente pares de elementos adjacentes e trocar de posição se houver um erro de ordem.

Além disso, na classificação por bolhas, a troca de elementos não usa armazenamento temporário, minimizando assim os requisitos de espaço. A principal desvantagem é seu baixo desempenho quando a lista contém muitos itens. A ordenação por inserção é um algoritmo de ordenação que, dada uma lista de arrays, constrói um array final, um elemento por vez, um elemento por vez. Assim como o algoritmo de ordenação quadrática, é muito eficaz para problemas com pequenas entradas e é o mais eficiente desse algoritmo de ordenação.

A principal vantagem da ordenação por inserção é sua simplicidade, além de ter bom desempenho em listas pequenas. É um algoritmo de classificação de localização e, portanto, possui requisitos mínimos de espaço.. A desvantagem é que seu desempenho não é tão bom quanto outros algoritmos de ordenação. A classificação por inserção também não é adequada para listas grandes devido às muitas etapas necessárias. O Quicksort é considerado por alguns como o melhor algoritmo de ordenação porque possui vantagens significativas de eficiência, pois funciona bem para grandes listas de itens.

Com a classificação no local, não é necessário espaço de armazenamento adicional. Uma pequena desvantagem que apresenta é que seu pior desempenho é semelhante ao desempenho médio dos demais algoritmos mencionados acima. No entanto, é importante observar que esse cenário de pior caso é muito raro. De maneira mais geral, o quicksort produz o método de classificação mais eficiente e amplamente utilizado para classificar listas de qualquer tamanho. A classificação de mesclagem é um exemplo de um algoritmo de classificação de comparação de divisão e conquista.

A idéia básica é decompor o problema em vários subproblemas, resolver esses subproblemas recursivamente e conquistar (após todos os subproblemas serem resolvidos, conquistar, ou seja, a união de problemas de subproblemas. o consumo é alto e o tempo de execução é alto, tornando esta técnica ineficiente para alguns problemas.Como desvantagem, tem sobrecarga de memória adicional.O algoritmo cria uma cópia do vetor para cada nível de chamadas recursivas.

Shell Sort, criado por Donald Shell, é o algoritmo de ordenação mais eficiente de todos. É uma melhoria em relação ao método inline, o algoritmo difere do método inline na medida em que ao invés de tratar o array a ser ordenado como um único segmento, ele considera vários segmentos, aplicando o método inline a cada segmento. Shellsort é uma excelente escolha para arquivos de tamanho médio. Sua implementação é simples e requer apenas um pouco de código.

**Questão extra 3) 1.5 ponto**

Em relação aos métodos de busca, descreva sucintamente cada um deles (3 linhas por método), falando de suas vantagens e desvantagens e como funciona cada método. São 5 métodos para descrever (busca sequencial, binária, tabela hash, busca em largura e busca em profundidade).

A busca sequencial é uma busca simples, a mais simples, ela compara a chave e o valor de cada elemento do array através de uma lista, então se um dos elementos buscados for igual, a posição correspondente à chave é retornada, então sua busca o tempo será bastante reduzido, porém, como desvantagem, a tabela deve ser ordenada.

Binário é um algoritmo eficiente para encontrar um item em uma lista ordenada, dividindo iterativamente pela metade o conteúdo da lista que deve conter o item, até reduzir o número de lugares que o item pode existir, até reduzir a apenas um. A vantagem é que a busca é eficiente e fácil de implementar, mas a desvantagem é que nem todas as permutações são ordenadas.

Uma tabela de hash tem uma chave de pesquisa que é convertida em um endereço. Deve haver uma maneira de lidar com o conflito, pois ele se traduz no mesmo endereço da tabela. Como o conjunto de teclas é pequeno, podemos atribuir a cada tecla uma tabela de posições. Exemplo |KEY | = |SET|, que tem as vantagens de implementação simples e alta eficiência de custo de busca, mas também traz o problema de precisar ordenar toda a tabela para recuperar registros ordenados por chave.

É um algoritmo de busca usado para realizar uma busca, obviamente em um dendrograma, primeiro você explora todos os seus vizinhos a partir do vértice raiz, depois, para chegar a um vértice inexplorado, exploramos todos os seus vizinhos para chegar ao mesmo, até encontrar seu alvo de pesquisa. Suas vantagens são o acesso rápido e direto, mas suas desvantagens são a ineficiência de armazenamento e pode haver muito ou pouco espaço.

Algoritmo que realiza uma busca, disponível para grafos e árvores, partindo da raiz, ou seja, escolhendo um nó como raiz, e se estiver no grafo, explorando as possibilidades de cada um de seus ramos antes de retornar. Ele se aprofunda cada vez mais na ignorância expandindo seus nós filhos.