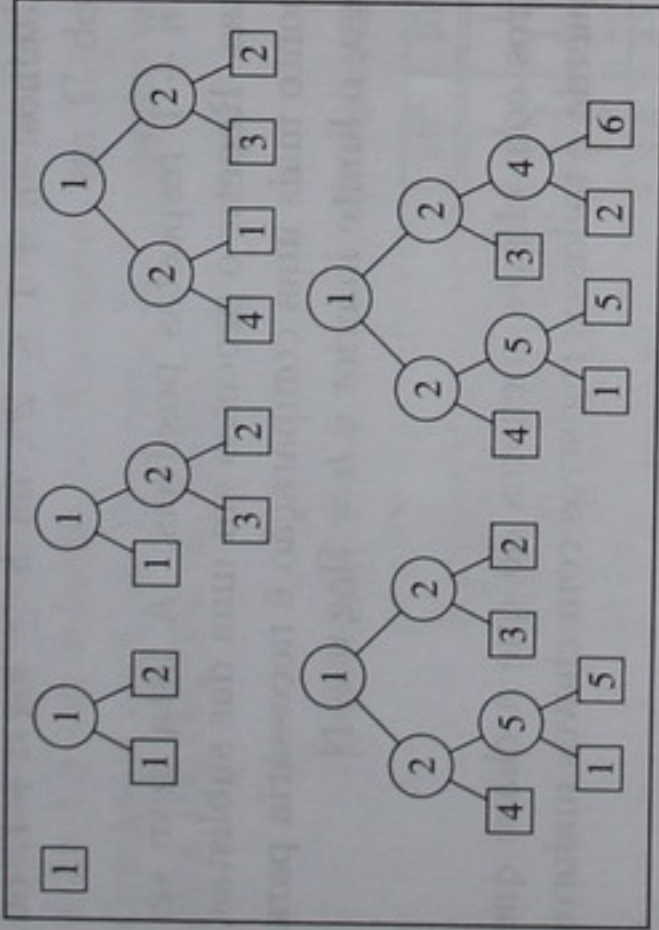


5.24.



Capítulo 6

6.1.

a) Em um ambiente de memória virtual devemos escolher algoritmos que possuam uma **localidade de referência espacial** pequena, isto é, cada referência à memória tem probabilidade alta de ocorrer em uma área relativamente próxima a outras áreas recentemente referenciadas, o que significa que a necessidade de transferir dados da memória externa para a interna é pouco frequente. Como exemplo, o *Quicksort* tem duas localidades de referência, pois a maioria das referências a dados ocorre em um dos dois apontadores utilizados na partição do arquivo. O algoritmo de Inserção deve funcionar razoavelmente, pois um registro é retirado da sequência origem e colocado no lugar apropriado na sequência destino, provocando boa localidade de referência (lembrar, entretanto, que o algoritmo é $O(n^2)$).

b) Sim, melhora porque pelo menos o nó raiz e provavelmente a maioria das páginas filhas da raiz estarão residentes na memória principal todo o tempo.

6.2.

a) **Localidade de referência espacial** é uma propriedade exibida por programas em relação aos seus dados em que cada referência a uma localidade de memória tem grande chance de ocorrer em uma área que é relativamente próxima de outras áreas que foram recentemente referenciadas.

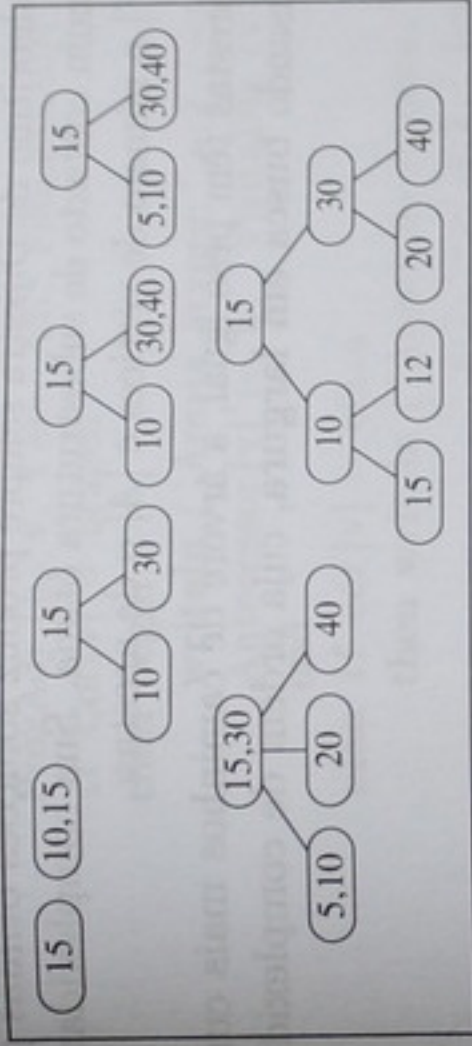
b) A distância entre acessos é a métrica mais tradicional de medição de localidade de referência espacial. Essa métrica pode ser aplicada a contextos mais específicos, como uma dada estrutura de dados.

c) **Localidade de referência temporal** é uma propriedade exibida por programas em relação aos seus dados que expressa a probabilidade de um dado ser acessado novamente considerando o momento do seu último acesso.

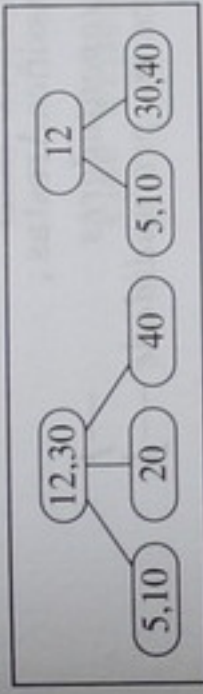
d) A distância de pilha dos acessos é a métrica mais tradicional de medição de localidade de referência temporal. Essa métrica pode ser aplicada a contextos mais específicos, como uma dada estrutura de dados.

6.3.

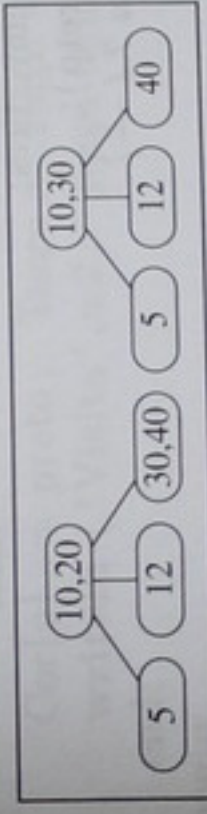
a)



b)



ou



Capítulo 7

7.3.

a) O problema é modelado como um grafo não direcionado ponderado, onde vértices representam as posições onde máquinas deverão ser instaladas e arestas representam conexões diretas entre elas. O peso de cada aresta representa a distância entre elas, ou seja, o tamanho total do cabo necessário para conectá-las diretamente. O que se busca é o custo total de uma **árvore geradora mínima** do grafo.

b) Devemos utilizar o **algoritmo de Prim** ou o **algoritmo de Kruskal** para obter a árvore geradora mínima, modificando-o de forma a contabilizar o peso de cada aresta adicionada a árvore em um contador do peso total. Ao final, basta multiplicar esse contador por C . Tanto o algoritmo de Kruskal quanto o de Prim são ótimos. A modificação realizada não altera a otimalidade pois não afeta as escolhas de arestas.

c) O problema pode ser modelado como um grafo não direcionado, onde os vértices representam as posições onde máquinas deverão ser instaladas e arestas