1. Aplique o algoritmo *DistribuitionCounting sorting* para b,c,d,c,b,a,a,b assumindo que a lista de valores possível é [a, b, c, d]

2. Aplique *Horspool* para buscar BAOBAO em BESS_KNEW_ABOUT_BAOBABS

3. Considere o problema de buscar por genes em sequencias de DNA usando Horspool. Uma sequencia de DNA é representada por um texto no alfabeto [A, C, G, T].
(a) construa a shift table para a sequencia TCCTATTCTT
(b) aplique Horspool para localizar a sequencia do item (a) na sequencia de DNA TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT

4. Para a entrada 30, 20, 56, 75, 31, 19 e função hash h(K)=k mod 11
(a) construa a open hash table
(b) encontre o número máximo de comparações para uma busca com sucesso nessa tabela

5. Para a entrada 30, 20, 56, 75, 31, 19 e função hash h(K)=k mod 11
(a) construa a closed hash table
(b) encontre o número máximo de comparações para uma busca com sucesso nessa tabela

6. Seja uma árvore B cuja raiz pode possuir até 3 chaves.
(a) qual a configuração da árvore após a inserção, em uma árvore vazia, das chaves:
4 7 10 11 14 12 16 15 19 24 34 25 28 20 43 40 46 60 68 80 51 75 22 9 63 77 21 5 13 44
(b) dê um exemplo, na árvore gerada acima, de uma chave, eliminada, causaria alteração na altura da árvore se fosse utilizado o algoritmo original de eliminação de chave
(c) como a alteração na altura, indicada na passo anterior, pode ser evitada?

7. Construa uma tabela que indica qual seria o número de acessos a disco (i.e., proporcional à altura da árvore), para valores de ordem da árvore m=64, 128, 256, 512, 1024 e 2048, e número de registros igual a 1 mil, 1 milhão, 1 bilhão, 1 trilhão e 1 quatrilhão. Desenhe o gráfico correspondente.

8.
(a) Calcule C(6,3) utilizando o algoritmo de PD.
(b) Calcule C(6,3) utilizando C(n,k) = n! / (k!(n-k)!)
(c) Discuta o uso das abordagens (a) e (b) para n e k quaisquer.

9. Aplique Warshall a um dígrafo conexo com 5 nós e 5 arestas de sua escolhe, sendo que arestas conectam (i,j) para $i \neq j$.

10. Calcule complexidade de tempo do Warshall.

11. Aplique Floyd-Warshall a um dígrafo conexo com 5 nós e 8 arestas de sua escolha, sendo que arestas conectam (i,j) para i≠j, e arestas possuem pesos com valores inteiros entre 2 e 5.

12. Calcule complexidade de tempo do Floyd-Warshall.

13. Explique a relação de recorrência utilizada no algoritmo OptimalBST()

14. Reescreva o algoritmo OptimalBST() para reduzir o tempo de execução para $\Theta(n^2)$

15. Explique a relação de recorrência utilizada no algoritmo OptimalBST()

16. Explique a relação de recorrência utilizada no algoritmo MFKnapsack()

17. Aplique o algoritmo MFKnapsack() *com memória* para um conjunto de 5 itens, escolha o valor e o peso de cada item, e escolha uma capacidade de mochila que comporte 3 dos itens.

18. Aplique o algoritmo Prim(G) para um grafo conexo com 5 nós e 8 arestas de sua escolha, sendo que arestas conectam (i,j) para i≠j, e arestas possuem pesos com valores inteiros entre 2 e 5.

19. Discuta por que Prim leva a uma solução ótima.

20. Calcule a complexidade de tempo de Prim.

21. Aplique o algoritmo Kruskal(G) para o mesmo grafo usado para Prim.

22. Discuta por que o Kruskal leva a uma solução ótima.

23. Calcule a complexidade de tempo de Kruskal.

24. Aplique o algoritmo Dijkstra() para dígrafo conexo com 5 nós e 8 arestas de sua escolha, sendo que arestas conectam (i,j) para i≠j, e arestas possuem pesos com valores inteiros entre 2 e 5.

25. Discuta por que o Dijkstra leva a uma solução ótima.

26. Calcule a complexidade de tempo de Dijkstra.

27. Construa o código de Huffman() para um alfabeto de 6 caracteres de sua escolha, sendo que todos caracteres possuem probabilidade diferentes.

28. Calcule a taxa de compressão obtida com o código produzido na questão anterior.