

**3.1**

1a. Give an example of an algorithm that should not be considered an application of the brute-force approach.

-> The algorithm for finding the binary representation of an integer.

5. Determine which of the three topologies, if any, the matrix represents. Design a brute-force algorithm for this task and indicate its time efficiency class.

-> For a ring topology, each vertex has two edges, thus each row in the matrix should contain 1 twice. A brute force algorithm would start by looking through row 0 to see if it has exactly two 1s. If this is not the case, the algorithm stops and we know the matrix is not representing a ring topology. Else, we move onto the next row in the matrix and continue to check each row until we either find a row without exactly two 1s or reach the end of the matrix. If we reach the end of the matrix, we know that it is the adjacency matrix of a graph with a ring topology. Since this algorithm will check every index of the matrix, it has a time efficiency of O(n2).

-> For a star topology, each vertex will have one edge except for one vertex which will have n-1 edges. A brute force algorithm would again need to look through each row at a time to see if there is either exactly one 1 or n-1 1s. If neither of these conditions are met, the algorithm stops and we know that the matrix does not represent a star topology. Else, we continue through each row of the matrix and continue to check each row until we have found a row that does not contain exactly one 1 and does not contain exactly n-1 1s, or does contain n-1 1s but we have already found a row with n-1 1s. This would mean that there are multiple vertices with multiple edges, thus the matrix wouldn’t represent a star matrix. If we reach the end of the matrix, we know that it is the adjacency matrix of a graph with a star topology. Since this algorithm will check every index of the matrix, it has a time efficiency of O(n2).

-> For a fully connected mesh topology, each vertex wil have n-1 edges, thus each row in the matrix should contain n-1 1s. A brute force algorithm would start by looking through the first row in the matrix to see if there is exactly n-1 1s. If this is not the case, the algorithm stops and we know that the matrix is not representing a fully connected mesh topology. Else, we move on to the next row in the matrix and continue to check each row until either we find a row without n-1 1s or reach the end of the matrix. If we reach the end of the matrix, we know that it is the adjacency matrix of a graph with a fully connected mesh topology. Since this algorithm will need to check every index of the matrix, it has a time efficiency of O(n2).

8.Sort the list E, X, A, M, P, L, E in alphabetical order by selection sort.

-> [E][X][A][M][P][L][E]

-> [A][X][E][M][P][L][E]

-> [A][E][X][M][P][L][E]

-> [A][E][E][M][P][L][X]

-> [A][E][E][L][P][M][X]

-> [A][E][E][L][M][P][X]

-> [A][E][E][L][M][P][X]

**3.2**

1 a-b. Find the number of comparisons made by the sentinel version of sequential search

a. in the worst case.

Cworst(n) = n + 1

b. in the average case if the probability of a successful search is .

Carg(n) = (1\*p/n + … +n\*p/n) + (n+1)(1-p)

= (p/n) (n(n+1)/ 2) + (n+1)(1-p)

3. A firm wants to determine the highest floor of its n-story

headquarters from which a gadget can fall without breaking. The firm has

two identical gadgets to experiment with. If one of them gets broken, it

can not be repaired, and the experiment will have to be completed with

the remaining gadget. Design an algorithm in the best efficiency class you

can to solve this problem.

If the first gadget is dropped from floors until it causes the gadget to malfunction or no such floor is encountered in the sequence before the top floor of the building is reached. In the former case, if the floor that causes the malfunction is higher than and lower than . So, begin droppin the second gadget from , , and so on until a drop causes the gadget to malfunction, making the floor directly before the current one the floor in question. If no floor causes the gadget to fail, the floor must be higher than , the last floor tried in the sequence. Therefore floors can continue being examined from that point, and so on until a gadget failure occurs or the last floor of the building is reached. The number of times two gadgets are dropped cannot exceed making this experiment have a runtime of

5a-c. How many comparisons (both successful and unsuccessful) are made by the

brute-force string-matching algorithm in searching for each of the following

patterns in the binary text of 1000 zeros?

a. 00001

-> for 00001 there will be four successful comparisons and 1 unsuccessful comparisons on each trial before shifting to the next position in the pattern

000000

00001 // the 4 zeros will match but the 1 will not.

00001 // same thing here

The total comparisons will be 4980.

b. 10000

-> for 10000 there will be 1 unsuccessful comparison for each trial before shifting to the next position in the pattern.

000000

10000 // 1 will not match, the 4 zeros will

10000 // same thing here

The total comparisons will be 996.

c. 01010

-> for 01010 there will be 1 successful and one unsuccessful comparison for each trial before shifting to the next position in the pattern.

00000

01010 // 0 will match, 1 will not

01010 // same thing here

The total comparisons will be 1992.

**3.3**

1. Assuming that *sqrt* takes about ten times longer than each of the other operations in the innermost loop of *BruteForceClosestPoints*,which are assumed to take the same amount of time, estimate how much faster will the algorithm run after the improvement discussed in Section 3.3.

2+3+2+10 / 2+3+2 = 17/7 = 2.43

3. Let be real numbers representing coordinates of villages located along a straight road. A post office needs to be built in one of these villages   
a. Design an efficient algorithm to find the post-office location minimizing the average distance between the villages and the post office.

M <- n/2

Return Xm

b. Design an efficient algorithm to find the post-office location minimizing the maximum distance from a village to the post office.

M <- (X1 + Xn) / 2

I <- 1

While Xi < m do

I <-i +1

If Xi - X1 < Xn -Xi-1

Return Xi

Else return Xi-1

**3.4**

1. a. assuming that each tour can be generated in constant time, what will be the efficiency class of the exhaustive-search algorithm outlined in the text for the traveling salesman problem?

θ(n!) as n\*½(n-1)! is in the same growth order

b. If this algorithm is programmed on a computer that makes 10 billion additions per second, estimate the maximum number of cities for which the problem can be solved in

(i) one hour: 16! Is the largest we can get without going over 3.6e13

(ii) 24-hours. 17! Is the largest we can get without going over 8.62e14

(iii) one year. 19! Is the largest we can get without going over 3.1536E17

(iv) one century. 20! Is the largest we can get without going over 3.15576E19

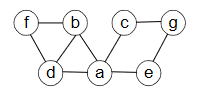
5. Give an example of the assignment problem whose optimal solution does not include the smallest element of its cost matrix.

The simplest assignment problem whose optimal solution does not include the smallest element of its cost matrix

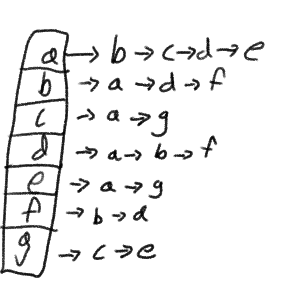
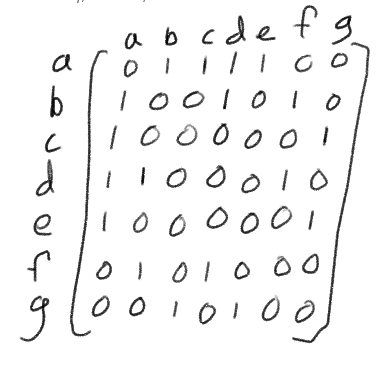
| 1 | 2 |
| --- | --- |
| 2 | 9 |

**3.5**

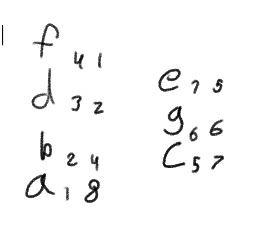
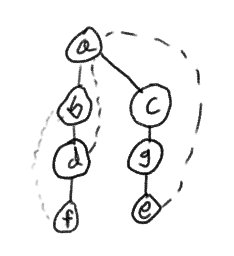
1. Consider the following graph.



a. Write down the adjacency matrix and adjacency lists specifying this graph. (Assume that the matrix rows and columns and vertices in the adjacency lists follow in the alphabetical order of the vertex labels.)



b. Starting at vertex and resolving ties by the vertex alphabetical order,traverse the graph by depth-first search and construct the corresponding depth-first search tree. Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in which the vertices became dead ends (popped off the stack).



(first number is order pushed, second is order popped)

2. If we define sparse graphs as graphs for which . Which implementation of DFS will have a better time efficiency for such graphs,the one that uses the adjacency matrix or the one that uses the adjacency lists?

Efficiency for adjacency matrix is and the efficiency for adjacency lists is . So if then the adjacency list would be more efficient with ) compared to the adjacency matrix’s efficiency of .

4. Traverse the graph of Problem 1 by breadth-first search and construct the corresponding breadth-first search tree. Start the traversal at vertex and resolve ties by the vertex alphabetical order.

