Design and Analysis of Algorithms Lab
Academic Year: 2020 - 21

Dr. Praveen Kumar Alapati
Sri. G. Brahmaiah (Ph.D. Scholar)
Brahmaiah20pcse001@mahindrauniversity.edu.in
praveenkumar.alapati@mahindrauniversity.edu.in

Department of Computer Science and Engineering

Ecole Centrale School of Engineering

DAA Lab 1 Due Date: February 7, 2021

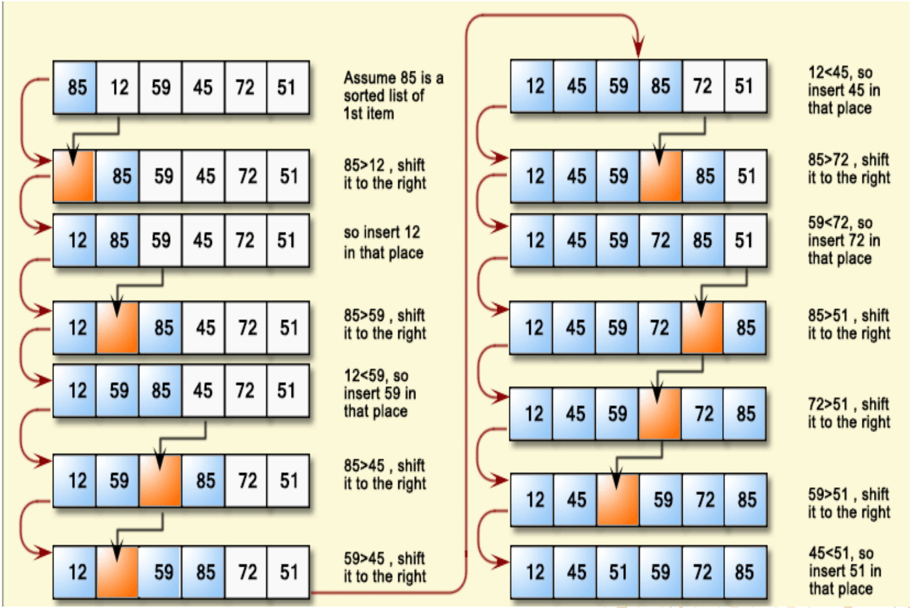
1. Use Selection Sort and Insertion Sort techniques to sort a set of student records by considering a specified field (Hall Ticket Number, Name, or Team Number).
2. Use Selection Sort and Insertion Sort techniques to sort a set of student records by considering all the fields in a specific order (Team Number, Hall Ticket Number, and Name).

Note:

- Input should be read from a file **DAALab_input1.txt**
- Output should be written into a file **DAALab_output1.txt**

Logic: Selection Sort

Logic: Insertion Sort



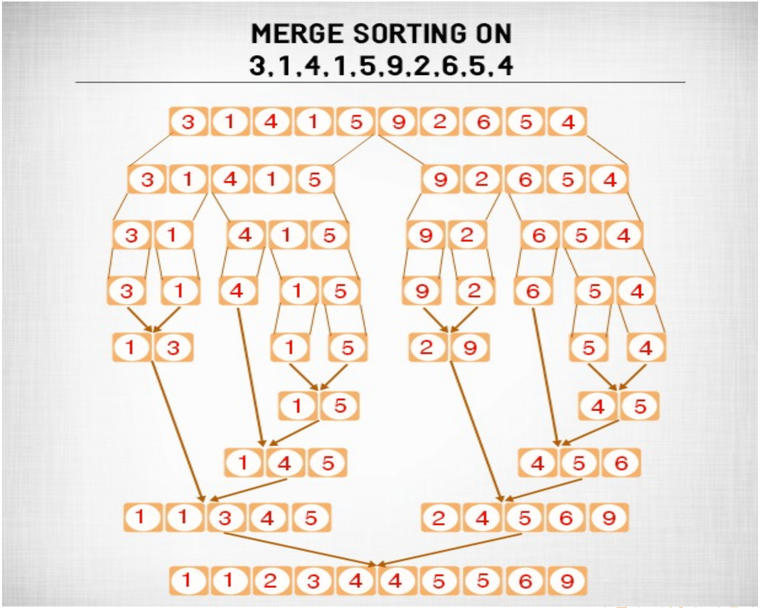
DAA Lab 2 Due Date: February 14, 2021

1. Use Merge Sort and Quick Sort techniques to sort a set of student records by considering a specified field (Hall Ticket Number, Name, or Team Number).
2. Use Merge Sort and Quick Sort techniques to sort a set of student records by considering all the fields in a specific order (Team Number, Hall Ticket Number, and Name).

Note:

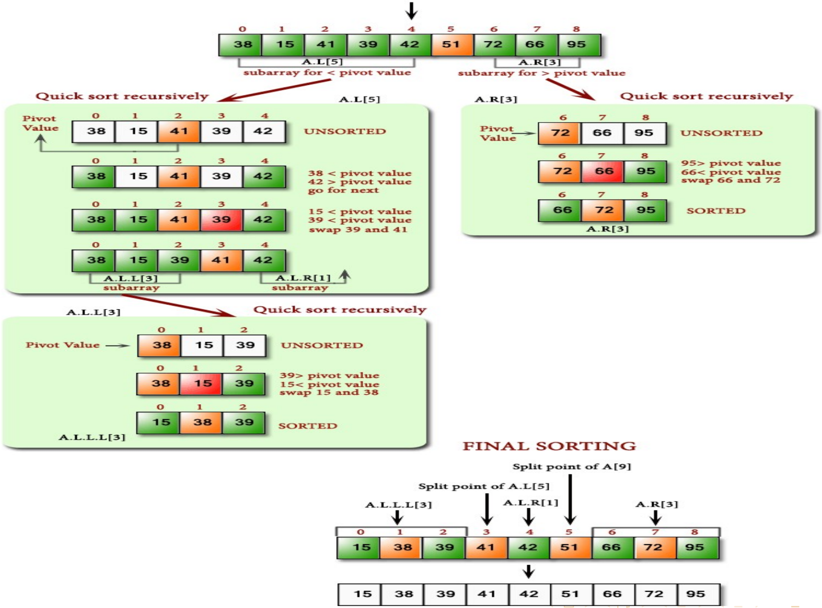
- Input should be read from a file **DAALab_input1.txt**
- Output should be written into a file **DAALab_output1.txt**

Logic: Merge Sort



Logic: Quick Sort

Logic: Quick Sort



DAA Lab 3 Due Date: February 21, 2021

1. Use Linear Search technique to search a student record by considering a specified field (Hall Ticket Number, Name, or Team Number).
2. Use Binary Search technique to search a student record by considering a specified field (Hall Ticket Number, Name, or Team Number).

Note:

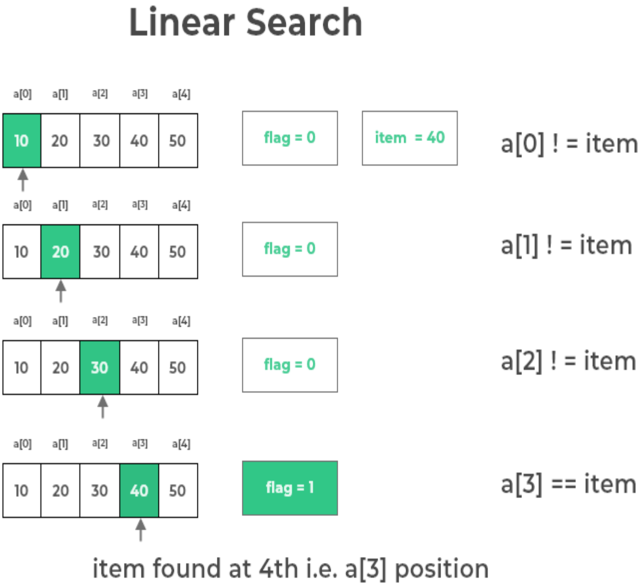
- Input should be read from a file **DAALab_input1.txt**
- Output should be written into a file **DAALab_output1.txt**

DAA Lab 3 Due Date: February 21, 2021

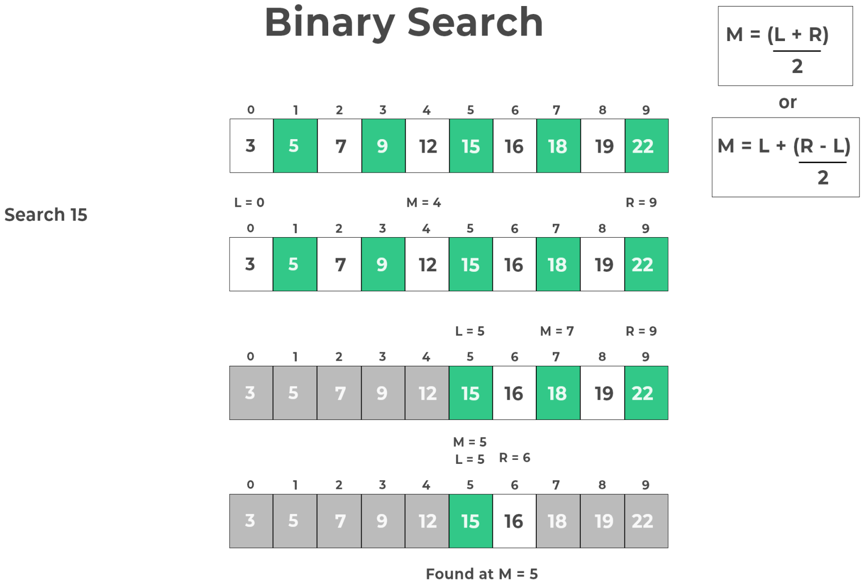
Bonus:

1. Use Fibonacci Search technique to search a student record by considering a specified field (Hall Ticket Number, Name, or Team Number).

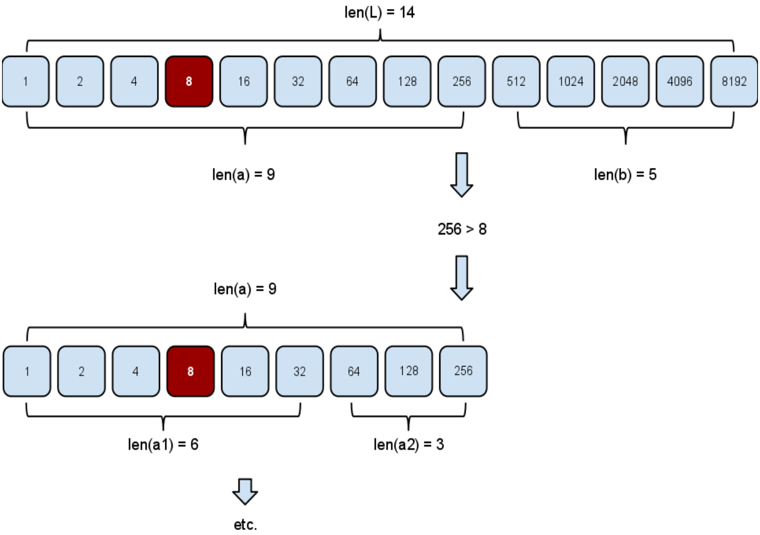
Logic: Linear Search



Logic: Binary Search



Logic: Fibonacci Search



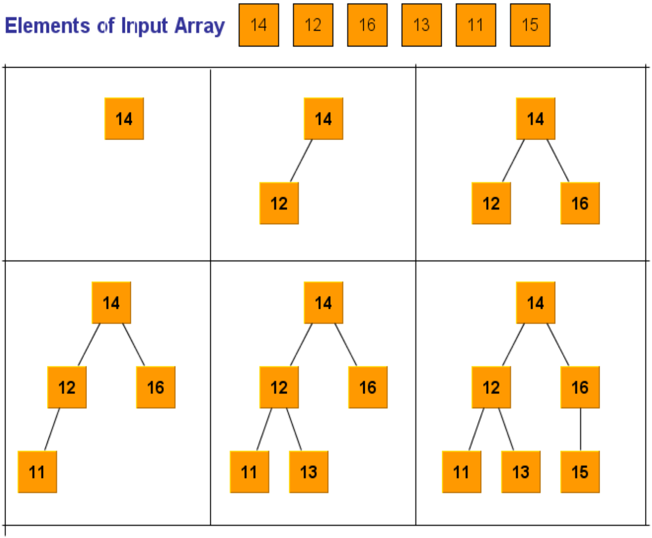
DAA Lab 4 Due Date: March 02, 2021

1. Use a Tree Sort technique to sort a set of student records by considering Hall Ticket Number.
2. Develop a program to multiply two square-matrices of order 1024 X 1024 using Block Matrix Multiplications by considering the block sizes: 4, 8, 16, 32, and 64. Use *gettimeofday*() for calculating *runtime* (the average of 5 runs). Draw a plot using *runtime* and block-size.

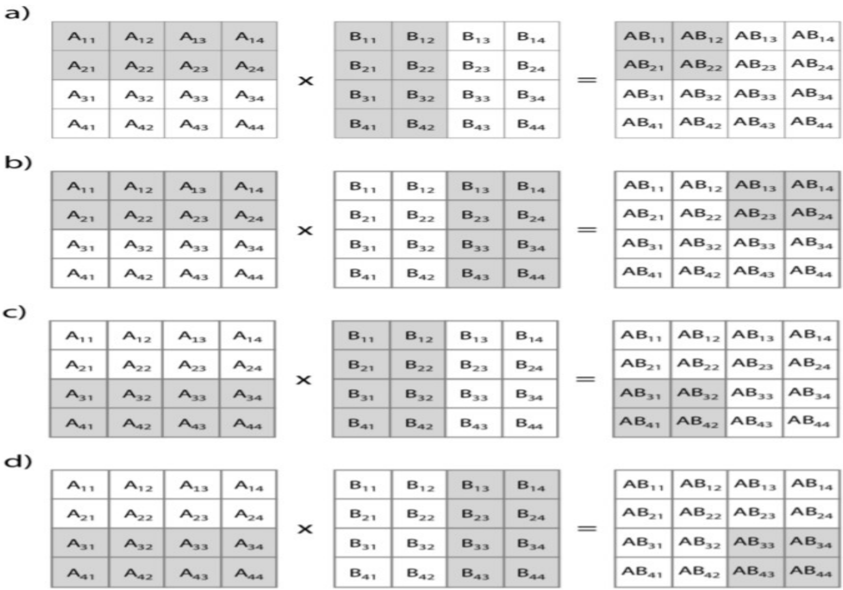
Note:

- Input should be read from a file **DAALab_input1.txt**
- Output should be written into a file **DAALab_output1.txt**

Logic: Tree Sort



Logic: Block Matrix Multiplication



DAA Lab 5 Due Date: March 14, 2021

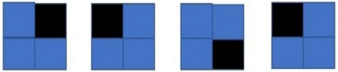
1. Develop a program for the Defective Chessboard problem (N=1024, 2048, and 4096). Use *gettimeofday*() for calculating *runtime* (the average of 5 runs).
2. Develop a program to multiply two square-matrices of order 1024 X 1024 using Strassen's Matrix Multiplication. Use *gettimeofday*() for calculating *runtime* (the average of 5 runs).

Bonus Problem Statements:

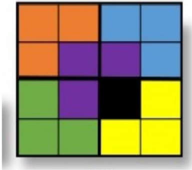
1. Given an array of n numbers and a positive integer i, write a program to find the i^{th} smallest element that runs in O(n) time.
2. Given two sorted arrays, each consisting of n numbers, write a program to find the median of $2n$ elements that runs in $\mathcal{O}(\log n)$ time.

Logic: Defective Chessboard

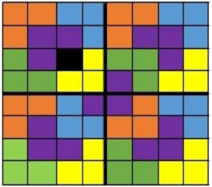
A chessboard that has one unavailable square. We have to cover the remaining squares using triominos.
(Triomino is an **L** shaped object and it is formed with three squares.)



2 x2 2 x2 2 x2 2 x2



4x4

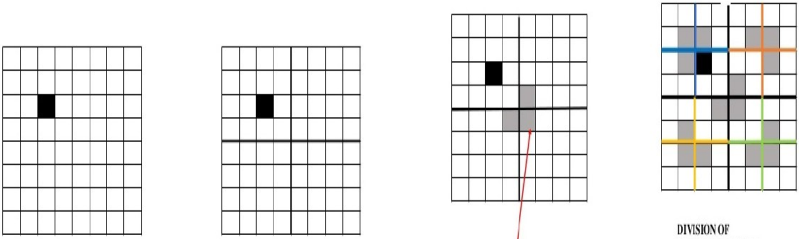


8x8

Black color square is the defective one.

Number of triomino's required for an $n \times n$ defective chess board: $\frac{n^2-1}{3}$.

8 X 8 Defective Chessboard



Creation of defective box

DIVISION OF PROBLEM INTO SUB PROBLEM

1. Divide the chessboard into 4 equal parts.
2. Identify the part which has the defective square and put a triomino that cover all the remaining three parts.
3. Now assume that all 4 parts are defective chessboards.
4. Repeat the steps 1 to 3 until all the squares are covered with triominos.

Defective Chessboard: Analysis

$$
\begin{aligned}
T(n) &= 4 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(1) \\
&= 4 \cdot T\left(\frac{n}{2}\right) + constant \\
&= 4 \cdot T\left(\frac{n}{2}\right) + constant \\
&= \Theta(n^2)
\end{aligned}
$$

Reasoning:

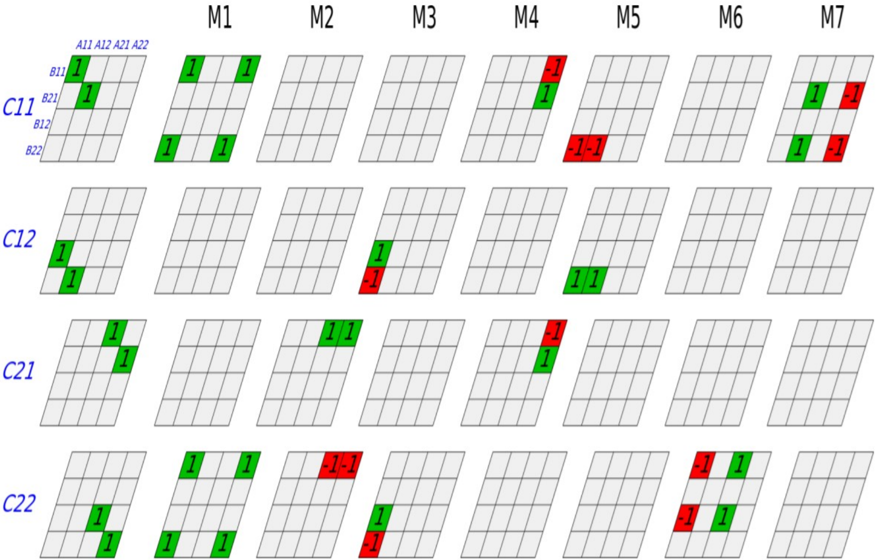
From case 1 of Master Theorem, where a=4, b=2, and f(n)= $\mathcal{O}(1)$
$n^{\log_b a} = n^{\log_2 4}$
$f(n) = n^{\log_2 4-\epsilon}$, where $\epsilon = 2$
So, $f(n)$ is polynomially less than $n^{\log_2 4} = n^2$.
$\therefore\ T(n) = \Theta(n^2)$

Logic: Strassen's Matrix Multiplication

$$
\begin{aligned}
M_1 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\
M_2 &= (A_{21} + A_{22}) \cdot B_{11} \\
M_3 &= A_{11} \cdot (B_{12} - B_{22}) \\
M_4 &= A_{22} \cdot (B_{21} - B_{11}) \\
M_5 &= (A_{11} + A_{12}) \cdot B_{22} \\
M_6 &= (A_{21} - A_{11}) \cdot (B_{11} + B_{12}) \\
M_7 &= (A_{12} - A_{22}) \cdot (B_{21} + B_{22})
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= M_1 + M_4 - M_5 + M_7 \\
C_{12} &= M_3 + M_5 \\
C_{21} &= M_2 + M_4 \\
C_{22} &= M_1 - M_2 + M_3 + M_6
\end{aligned}
$$

Strassen's Matrix Multiplication



Strassen's Matrix Multiplication:Analysis

$$
\begin{aligned}
T(n) &= 7 \cdot T\left(\frac{n}{2}\right) + 18 \cdot \mathcal{O}\left(\frac{n^2}{4}\right) \\
&= 7 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}\left(n^2\right) \\
&= 7 \cdot T\left(\frac{n}{2}\right) + c \cdot n^2 \\
&= \Theta(n^{2.81})
\end{aligned}
$$

Reasoning:

From case 1 of Master Theorem, where a=7, b=2, and f(n)= $\mathcal{O}(n^2)$
$n^{\log_b a} = n^{\log_2 7}$
$f(n) = n^{\log_2 7 - \epsilon}$, where $\epsilon = 0.81$
So, $f(n)$ is polynomially less than $n^{\log_2 7} = n^{2.81}$.
$\therefore \; T(n) = \Theta(n^{2.81})$

DAA Lab 6 Due Date: March 31, 2021

1. **Kanpsack Problem:** We are given with n objects and a knapsack with capacity M. Let w_1, w_2, w_3, ... w_n and p_1, p_2, ... p_n be the weights and profits of n objects, respectively. If we place a fraction $x_i, (0 \leq \ x_i \leq 1)$ of object i into the Knapsack, then we get a profit $p_i.x_i$ and kanpsack capacity is reduced by $M - w_i.x_i$. Write a program to find a solution vector (x_1,x_2,x_3, ... , x_n) in such a way that we have to get the maximum profit.
2. **Job Sequencing with Deadlines:** We are given with a machine and a set of n jobs. Each job i has an integer deadline (d_i) and a profit (p_i). Execution time of any job is one unit. If a job i is executed within its deadline, then we get profit p_i. Write a program to find a solution vector (x_1,x_2,x_3, ... , x_n) in such a way that we have to get the maximum profit.

An Example of Kanpsack Problem

Objects	1	2	3	4	5	6	7
Profit	10	5	15	7	6	18	3
Weight	2	3	5	7	1	4	1

M 15

An Example of Job Sequencing with Deadlines Problem

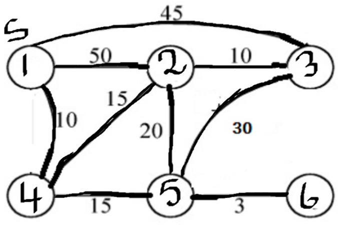
Job	Deadline	Profit
1	2	40
2	4	15
3	3	60
4	2	20
5	3	10
6	1	45
7	1	55

DAA Lab 7 Due Date: April 4, 2021

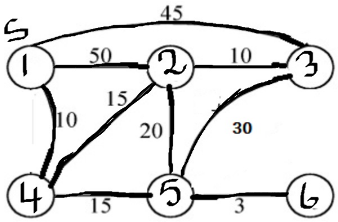
1. **Single Source Shortest Path (SSSP):** Given a connected weighted graph (weights represent the distances between two vertices), write a program to find a shortest path from a given source vertex 's' to every other vertex.
   Using the SSSP program find a shortest path between every pair of vertices.

2. **Huffman Coding:** Write a program to compress and decompress a file using a Huffman Coding. The uncompressed text file and the original text file should be the same.
   (**Size of orizinal file should be $\geq$ 1 MB**).

Single Source Shortest Path Problem

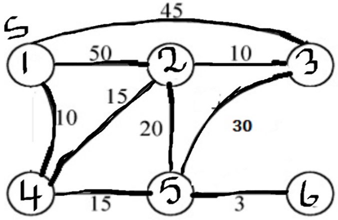


Single Source Shortest Path Problem

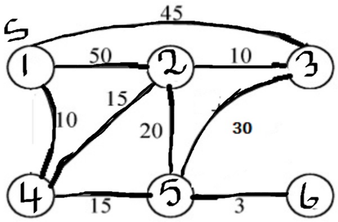


	Adjacency Matrix Representation					
	1	2	3	4	5	6
1	0	50	45	10	----	----
2	50	0	10	15	20	---
3	45	10	0	---	30	---
4	10	15	---	0	15	---
5	---	20	30	15	0	3
6	----	----	----	----	3	0

Single Source Shortest Path Problem



	1	2	3	4	5	6
1	-1	-1	-1	-1	-1	-1
2	-1	1	1	1	-1	-1
3	-1	4	1	1	4	-1
4	-1	4	2	1	4	-1
5	-1	4	2	1	4	5
6	-1	4	2	1	4	5



	1	2	3	4	5	6
1	-1	-1	-1	-1	-1	-1
2	-1	1	1	1	-1	-1
3	-1	4	1	1	4	-1
4	-1	4	1	1	4	---
5	-1	4	2	1	4	---
6	-1	4	2	1	4	5

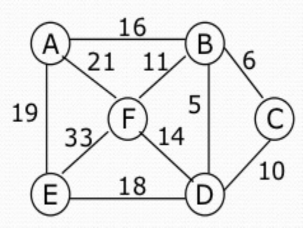
Single Source Shortest Path Problem: Algorithm

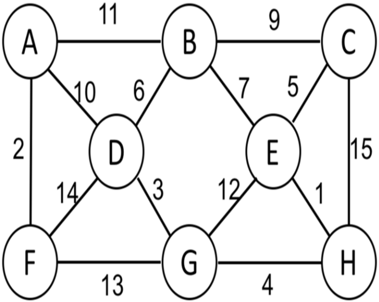
Algorithm 1: SSP(n, v_1, Cost[][], Dist[], Path[])

Result: Shortest Distances from source vertex to all other vertices

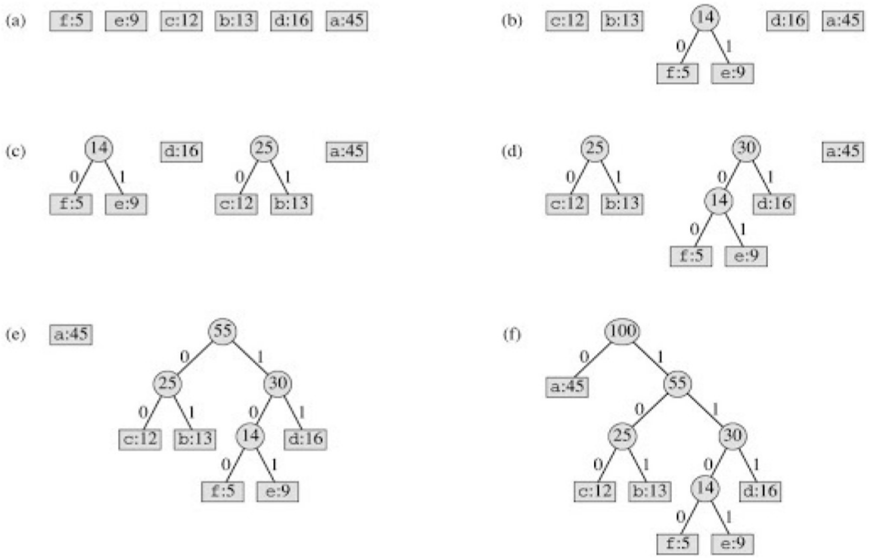
for $i = 1$ *to* $i \leq n$ **do**
  $S[i] = 0$; $Dist[i] = Cost[v_1][i]$; $Path[i] = -1$;
S[v_1]=1;
for $i = 2$ *to* $i \leq n$ **do**
  u=$ChooseMinimumDistanceVertex(from\ V - S)$;
  S[u]=1;
  for *each w adjacent to u and S[w]=0* **do**
    if *Dist[w]>Dist[u]+cost[u][v]* **then**
      $Dist[w] = Dist[u] + cost[u][v]$
      $Path[w] = u$

Find Shortest Paths from vertex A to all other vertices





Huffman Codes



DAA Lab 8 Due Date: May 9, 2021

1. **Travelling Salesperson Problem (TSP):** Given a connected weighted graph (weights represent the distances between two vertices), we have to find a tour with minimum distance (cost). Write a program to find an optimal tour.

2. **Reliability Design Problem:** Let us consider, we have to design an n-stage system with maximum reliability under the give cost constraints using device duplication technique. Write a program to identify the number of devices that can be connected in parallel.

$$f_n(C) = \max_{1 \le m_i \le u_i} \left\{ \phi_n(m_n) \,.\, f_{n-1}(C - c_n.m_n) \right)$$

Where m_n is the number of devices that can be connected in n^{th} Stage.
$\phi_n(m_n)$ is the reliability of Stage n.
Base case: $f_0(x) = 1$, where $x \geq 0$
$f_i(-ve) = 0$, where $0 \leq i \leq n$

DAA Lab 9 Due Date: May 17, 2021

1. **Matrix Chain Multiplication**: Given a chain of n matrices (i.e., A_1, A_2, A_3, ..., A_n) and dimensions (rows and columns) of the matrices are $p_0 \times p_1$, $p_1 \times p_2$, $p_2 \times p_3$ ... $p_{n-1} \times p_n$, respectively.
   Write a program to find an order(or parenthesize the matrices) to compute the product $A_1.A_2.A_3.$ ... $.A_n$ using minimum number of scalar multiplications.
2. **Longest Common Sub-Sequence (LCS)**:Let X_i=(x_1,x_2, ..., x_i) and Y_j= (y_1,y_2, ..., y_j) are two strings, then write a program to a Longest Common Sub-Sequence of X_i and Y_i.

Bonus: Write a program to find LCS of n strings. For example LCS of 4 strings { **aaabb**, **bbaaa**, **cccbb**, **bbccc**} is **bb**.

DAA Lab 10 Due Date: May 23, 2021

1. **N-Queens Problem:** Given an N $\times$ N chessboard and N-Queens. Place N-Queens on the chessboard in non-attackable positions. Consider the different values of N =8, 12, 16, and 20. Store all the solutions in a file.
2. **Sum of Subsets Problem:** We are given with n distinct positive numbers (usually called weights) and a value m. Write a program to find all the subsets of these n numbers whose sums are m (Use Backtracking).
3. **Graph Coloring Problem:** Let G be a graph with n vertices. Write a program to assign colors to the vertices of G (using minimum number of colors) in such a way that no two adjacent vertices have the same color.
4. **Develop programs for 3-Puzzle, 8-Puzzle and 15-Puzzle Problems:**

8-Puzzle

1	2	
4	5	3
7	8	6

Initial State

1	2	3
4	5	6
7	8	

Goal State

15-Puzzle

4	5	7	10
8	1	2	12
3	6	9	11
14	13	15	

Initial State

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Final State

DAA Lab Submission Guide Lines

- Mail-ID: cs203.daa.mec@gmail.com (Doubt Clarification).
- Submission Link will be shared.
- Late Submission (<=3-Days):50% weightage will be given.
- Write a readme file to understand your solutions.
- Submit source files only (C or JAVA).

Lab Weightage - 30%.

Lab Instructor: Sri. Brahmaiah G

DAA (Design and Analysis of Algorithms) Lab

Reference Books:

1. Introduction to Algorithms, 3rd edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekaran.
3. Algorithms, 4th edition, Robert Sedgewick.
4. Design and Analysis of Computer Algorithms, Aho, Ullman, and Hopcroft.

Web Resources:

1. Algorithms by Robert Sedgewik
2. Algorithms by Abdul Bari
3. MIT - Open Courseware Videos on Algorithms
4. Data Structures and Algorithms