



Department of Computer Science and Engineering
List of Algorithm Projects

1. Money Exchange machine

- a. Give an input and a menu like the ATM Machine.
- b. Show the notes(tk.) that has to be given in exchange
- c. In case of other currency, convert it to tk. And show notes
- d. Create a file for maintaining the notes residing in the machine and check if the amount available in the machine or not
- e. Give it a realistic look

2. Species Finding

Suppose you are a computer scientist who works with various species. You are very interested in finding new species and classify them by comparing with the DNA of existing species. You need to compare the new DNA with other DNAs and categorize the new one.

3. Sorting Algorithms

Suppose you have the responsibility to make a details list of the player of our National Cricket Team.

- Make a list of all players with their details (Name, Age, Date of Birth, ICC Rankings, #OfTests, #Of_One Day Internationals, #Of_Twenty20 Internationals).
- The list will be displayed in Alphabetical Order of their name.
- You can sort the list according to other features.
- You can add new players and also search, update information and delete details of any player.

4. Restaurant Management System

Write a program to manage Food order in a restaurant. In a restaurant, the food order contains food number, size, and quantity.

- The order is served in the form of first come first serve basis.

- Anyone can edit his order, and can cancel the order with a certain time.
- Your program will display total number of orders received, served and waiting at any point in time.
- At the end of the day, total bill can be shown.

5. ***Ticket Booking System***

Design a system for Automated Bus Ticket Booking system. People can see total seats, can book a seat, can edit his booking or delete his reservation.

6. ***Hotel Management System***

Design a Hotel Management System. User can see the available rooms, details of each room (Price, capacity), can see the services. User can book a room. Can update or delete his reservation.

7. ***Student Report Card System***

Your program performs following features:

- Create student report card record
- Display all students' report card record. Can sort the data by name and marks
- Display specific student's report card record
- Display all students' grade report
- Modify student's report card record
- Delete student record

8. ***Phonebook Application***

Create a Phone Book. The phone book displays detail phone information like your favorite cell phone.

- Add record
- View List in Ascending Order
- Search record
- Modify record
- Delete record

9. Route Planner

Suppose you want to travel Bangladesh. You made a list of popular Places. There are several routes to travel.

- a. Now **find all possible ways** from one place to another.
- b. Now find the **shortest path** from one place to another with the **cost**.

10. Comparison of different sorting algorithms

Suppose you have given a huge amount of data and you have to sort them in Both Ascending and Descending Order. But the condition is that, you have to do it in all of the five sorting Algorithms- **Selection Sort, Bubble sort, Insertion Sort, Merge Sort and Quick Sort** and find out which algorithm works faster.

- a. Show the comparative complexity of all the above sorting
- b. Show the runtime from the code(at least one)

11. Pre-requisite and eligibility checker

- a. Create a graph structure of your whole B.Sc. course plan with a link with the pre-requisite
- b. For a course input, show the total course hierarchy.
- c. For a list of completed course, show the next course to be completed
- d. For both, check eligibility and if not then suggest the course to be completed previously

12. Hide your Message

- a. Apply a login system
- b. In sender end, An input interface for the message from the user
- c. Apply Huffman coding for hiding and compressing the data and send the Huffman tree to receiver
- d. In receiver end, show the encrypted message and the Huffman tree
- e. Recover the original message

13. Subsequence Finder

Suppose you have some input sequences. Your task is to find the **increasing** and **decreasing** subsequences.

- a. You have to find out all the **Increasing** and **Decreasing** Subsequence from the Input Strings.

- b. Among them identify the **LIS/LDS** and its length for individual sequence.
- c. You can also **compare** between the LIS/LDS of each string.

14. Filling your Shopping Cart

You went to the market to buy some Items. You have a bag whose size is fixed. You can take as many quantity of any item as you want but you cannot exceed the limit of the bag.

Now find out, by taking which items you can make the maximum profit.

15. Car parking System

- a. Create a system for 10 parking lot
- b. Input a dynamic table of car weight and duration of residing
- c. Element in the input table can be removed and added dynamically
- d. Apply an (knapsack) algorithm to detect which car has to be given priority in parking
- e. Apply timer which will announce the waiting cars which are selected to get parking space.
- f. Give it an attractive look (with car image) like real life simulation

16. Blood Donation System

A system in which data of Patient, data of donor, data of blood bank would be saved and will be interrelation with each other

- **DATA OF PATIENT** – Patient Name, Patient Id, Patient Blood Group,
- **DATA OF DONAR** – Donar Name, Donar Id, Donar Blood Group, Donar Medical report, Donar Address , Donar Contact number
- **DATA OF BLOOD BANK** – Blood Bank Name, Blood Bank Address, Blood bank Donars name, Blood Bank Contact Number, Blood Bank Address.

What you need to know

- The Project is Group Based. **Maximum 4 members** in each group.
- Choose any project from the above list. You can also bring up **your own ideas**.
- **No two groups can have the same Project.**
- You have to prepare a **Google site** for the project presentation

- **Copying is Strictly Prohibited.**
- You will have to create a **Google site** and write a **report** on your project.

****** Content of the Report**

1. Introduction
2. Objective
3. Description
4. Requirement Specification
5. Sample Input and Output
6. Challenges
7. Conclusion