Activity 1

Bubble sort & Insertion sort

A bubble sort is performed on the following list:

3, 5, 8, 17, 12, 15, 18, 23, 1

1. Describe how a bubble sort works

Bubble sort works by bubbling up the largest number to the end of the array after each pass. It works by comparing two elements in an array and checking whether the next element is smaller and if it is, the lower number will swap with the larger number and change its index position within the array. This continues until the largest number is last.

2. What is the sequence after the first pass is completed?

3, 5, 8, 12, 15, 17, 18, 1, 23

3. How many passes through the list will be required to sort the items into ascending numerical sequence?

8

——————————

An insertion sort is performed on the following list:

3, 5, 8, 17, 12, 15, 18, 23, 1

1. Describe how an insertion sort works

An insertion sort is similar to bubble sort but it sorts in a opposite way. It drops the smallest element to the first index position in an array. This continues until the smallest number is first.

2. What is the sequence after the first pass is completed?

3, 5, 8, 17, 12, 15, 18, 23, 1 - The sequence remains unchanged as the smallest number is already at the first index, whereas if it wasn't then it would sort until the smallest number dropped to the first index position.

3. Why is an insertion sort quicker than a bubble sort?

The reasons insertion sort is faster than bubble sort is because there are fewer swaps as insertion moves elements instead of swapping every pair. Plus if the array is already almost sorted then it performs much faster.

Activity 2

Java Implementation

Find a Java implementation of an algorithm we discussed in this session (or a related one that interests you).
Try to get the Java code working in your IDE and use the debugger to follow its execution.

```
public class CollaborativeFiltering {

    public static void main(String[] args) {
        // User item ratings
        int[][] ratings = {
            {5, 4, 0, 0, 3}, // User 1
            {4, 0, 0, 2, 3}, // User 2
            {5, 4, 3, 0, 0}, // User 3
            {0, 0, 5, 3, 2}, // User 4
            {4, 3, 5, 0, 0}  // User 5
        };

        //Recommending an item for user 1
        int userIndex = 0;
        int recommendedItem = recommendItem(ratings, userIndex);

        System.out.println("Recommended item for user " +
(userIndex + 1) + ": Item " + (recommendedItem + 1));
    }
    public static int recommendItem(int[][] ratings, int 
userIndex) {
        //Find the most similar user to the given user
        int bestMatchUser = -1;
        int highestSimilarity = -1;

        for (int i = 0; i < ratings.length; i++) {
            if (i != userIndex) {
                int similarity = 
calculateSimilarity(ratings[userIndex], ratings[i]);
                if (similarity > highestSimilarity) {
                    highestSimilarity = similarity;
                    bestMatchUser = i;
                }
            }
        }

        //Recommend an item based on the most similar user
        for (int i = 0; i < ratings[bestMatchUser].length; i++) {
            if (ratings[userIndex][i] == 0 && 
ratings[bestMatchUser][i] > 0) {
                return i; // Recommend the first item that the 
best match user liked
            }
        }
        return -1; //If no recommendation, return -1
    }
```

——

Activity 3

An interesting Algorithm

Collaborative filtering

What is interesting or inspiring about the algorithm(s)?

Collaborative filtering ability to improve accuracy by itself and learn from user interactions and behaviours fascinates me as it is able to make recommendations without knowing the user or understanding the content it is recommending.

Has it evolved (e.g., been optimised) over time?

Collaborative filtering started off with basic approaches in the 1990s, where it was used to compare user ratings. Eventually this grew into user based filtering, where recommendations were made based on similar user behaviour. As the algorithm gained in popularity, due to how versatile and useful the algorithm could be, it started to be adopted by big companies such as Amazon, who introduced item based filtering . Then the matrix factorisation and the netflix prize competition advanced research into improving recommendation accuracy which led to advancements in machine learning techniques which led to what is used in the present day such as NCF, deep learning models, Context-Aware and Knowledge-Based Systems etc. To this day the algorithm is still evolving and there are always new innovations being made in this area.