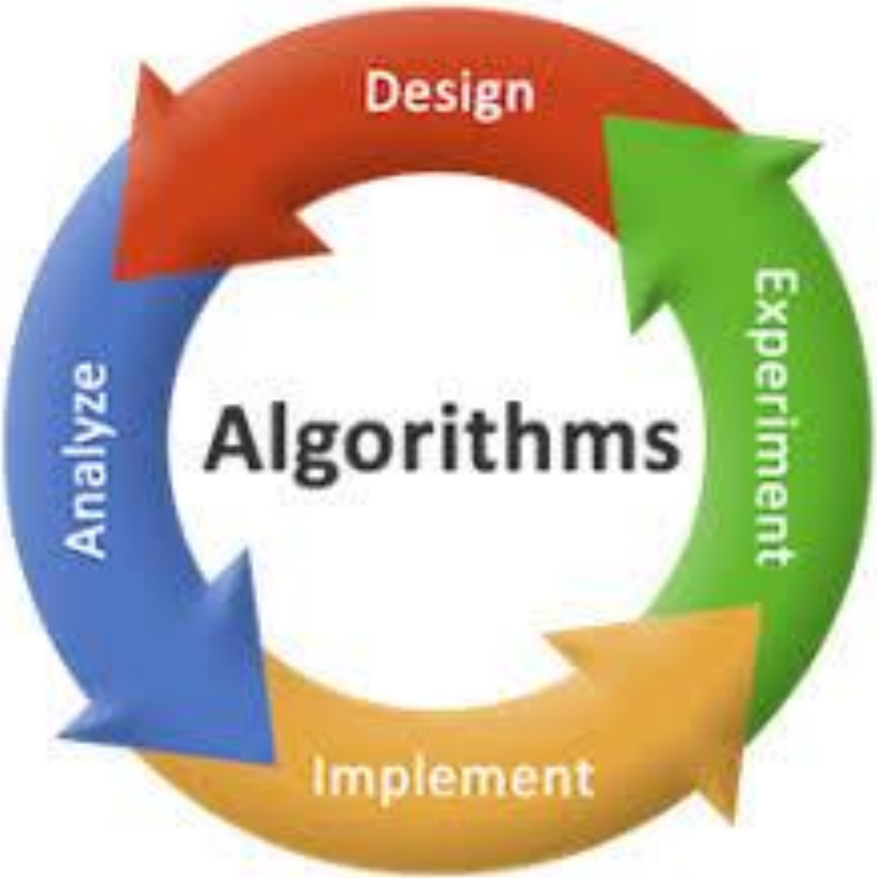

Design & Analysis of Algorithms

Lecture 13

Brute Force-Assignment Problem

Assignment Problem

Assignment Problem

- There are ***n*** people (agents) who need to be assigned to execute ***n*** tasks, one person per job. (That is, each person is assigned to exactly one job and each job is assigned to exactly one person.)
- The cost that would accrue if the **i^{th}** person is assigned to the **j^{th}** job is a known quantity **C[i,j]** for each pair i,j = 1, 2, …, n.
- The problem is to find an **assignment** with the **minimum total cost**.

Assignment Problem

Example

- A taxi firm (**Ola/Uber**) has three taxis (agents) available, and three customers (tasks) wishing to be picked up as soon as possible.
- The firm prides itself on speedy pickups, so for each taxi the "**cost**" of picking up a particular customer will depend on the time taken for the taxi to reach the pickup point.
- The solution to the assignment problem will be whichever combination of taxis and customers results in the **least total cost**.

Assignment Problem

- A small instance of this problem follows, with the table entries representing the assignment costs **C[i,j]**:

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

Assignment Problem

- It is easy to see that an instance of the assignment problem is completely specified by its **cost matrix** **C**.
- In terms of this matrix, the problem is to select one element in each row of the matrix so that all selected elements are in different columns and the total sum of the selected elements is the smallest possible.
- **Note** that no obvious strategy for finding a solution works here.

Assignment Problem

- **For example**, we cannot select the smallest element in each row, because the smallest elements may happen to be in the same column.
- In fact, the smallest element in the entire matrix need not be a component of an optimal solution.
- Thus, opting for the exhaustive search may appear as an unavoidable evil.

Assignment Problem

Brute-Force Approach

- We can describe feasible solutions to the assignment problem as ***n***-tuples $<j_1, \ldots, j_n>$ in which the i^{th} component, i = 1, …, n, indicates the column of the element selected in the i^{th} row (i.e., the job number assigned to the i^{th} person).

Assignment Problem

Brute-Force Approach

- **For example**, for the cost matrix (below), <2,3,4,1> indicates: **???**
- the assignment of Person 1 to Job 2, Person 2 to Job 3, Person 3 to Job 4, and Person 4 to Job 1.

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

Assignment Problem

Brute-Force Approach

- There should be a **one-to-one** correspondence between **feasible assignments** and **permutations** of the first **n** **integers**.

1. generating all the permutations of integers 1,2,…, n;
2. computing the total cost of each assignment by summing up the corresponding elements of the cost matrix;
3. finally selecting the one with the smallest sum.

Assignment Problem

Brute-Force Approach

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

$$C = \begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix}$$

$\langle 1, 2, 3, 4 \rangle$ cost = 9 + 4 + 1 + 4 = 18

$\langle 1, 2, 4, 3 \rangle$ cost = 9 + 4 + 8 + 9 = 30

$\langle 1, 3, 2, 4 \rangle$ cost = 9 + 3 + 8 + 4 = 24

$\langle 1, 3, 4, 2 \rangle$ cost = 9 + 3 + 8 + 6 = 26

$\langle 1, 4, 2, 3 \rangle$ cost = 9 + 7 + 8 + 9 = 33

$\langle 1, 4, 3, 2 \rangle$ cost = 9 + 7 + 1 + 6 = 23

etc.

Assignment Problem

Brute-Force Approach

- Since the number of permutations to be considered for the general case of the assignment problem is

$$n!$$

- exhaustive search is impractical for all but very small instances of the problem.
- Fortunately, there is a much more efficient algorithm for this problem called the **Hungarian method.**

References

- **Chapter 3**: Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Pearson Education, Third Edition, 2017.

Homework

- Eight-queens problem Consider the classic puzzle of placing eight queens on an 8×8 chessboard so that no two queens are in the same row or in the same column or on the same diagonal.
- Magic squares A magic square of order n is an arrangement of the integers from 1 to n^2 in an $n \times n$ matrix, with each number occurring exactly once, so that each row, each column, and each main diagonal has the same sum.