43159

ASSIGNMENT - 4

PROBLEM STATEMENT: Write a program to solve assignment problem

THEORY:

Lets say we have N workers & N machines. Any worker can be assigned any machine, incurring some efficiency that may vary depending on machine assignment. It is required to perform all jobs by assigning exactly 1 worker to each machine & exactly 1 job to each agent in such a way that total cost of assignment is minimized, else overall efficiency is improved.

ex:

	J_1	J_2	J_3	J_4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

- We try to calculate cost when J_2 is assigned to A.
- Since J_2 is assigned to A, cost becomes 2 & J_2 & A become unavailable.
- Now we try to assign J_3 to B as it has min. cost from list of unassigned tasks jobs. Cost becomes 2 + 3 = 5 & J_3 & B become unavailable

43159

- Finally J_1 gets assigned to c as it has min. cost among unassigned tasks & J_4 gets assigned to d as it is the only job left.

Total cost becomes = 2 + 3 + 5 + 4 = 14

- ALGORITHM :

findMinCost() uses least() & add() to maintain the list of live nodes, least() finds a live node with least cost, deletes it from the list & returns it, add(x) calculates cost of of x & adds it to the list of live nodes

```
node {
        int job_no.;
        int worker_no;
        int parent;
        int cost;
}

findMinCost ( costMatrix mat[][])
{
    Initialise list of live nodes
    with root of search tree, i.e., Dummy node

    while( true) {
        E = least()
        if (E is a leaf node) {
                printSolution()
                return
        }
```

43159

```
        for each child x of E
        {
            add(x);
            x → parent = E;
        }
    }
}
```

Solved Example:

```
int CostMatrix[N][N] =
  {
     9,  2,  7,  8
     6,  4,  3,  7
     5,  8,  1,  8
     7,  6,  9,  4
  }
```

O/P: A : 1
B : 0
C : 2
D : 3

CONCLUSION:
Thus, we have successfully implemented assignment problem.