SNS COLLEGE OF ENGINEERING

Coimbatore-107

COURSE NAME: ANALYSIS OF ALGORITHM

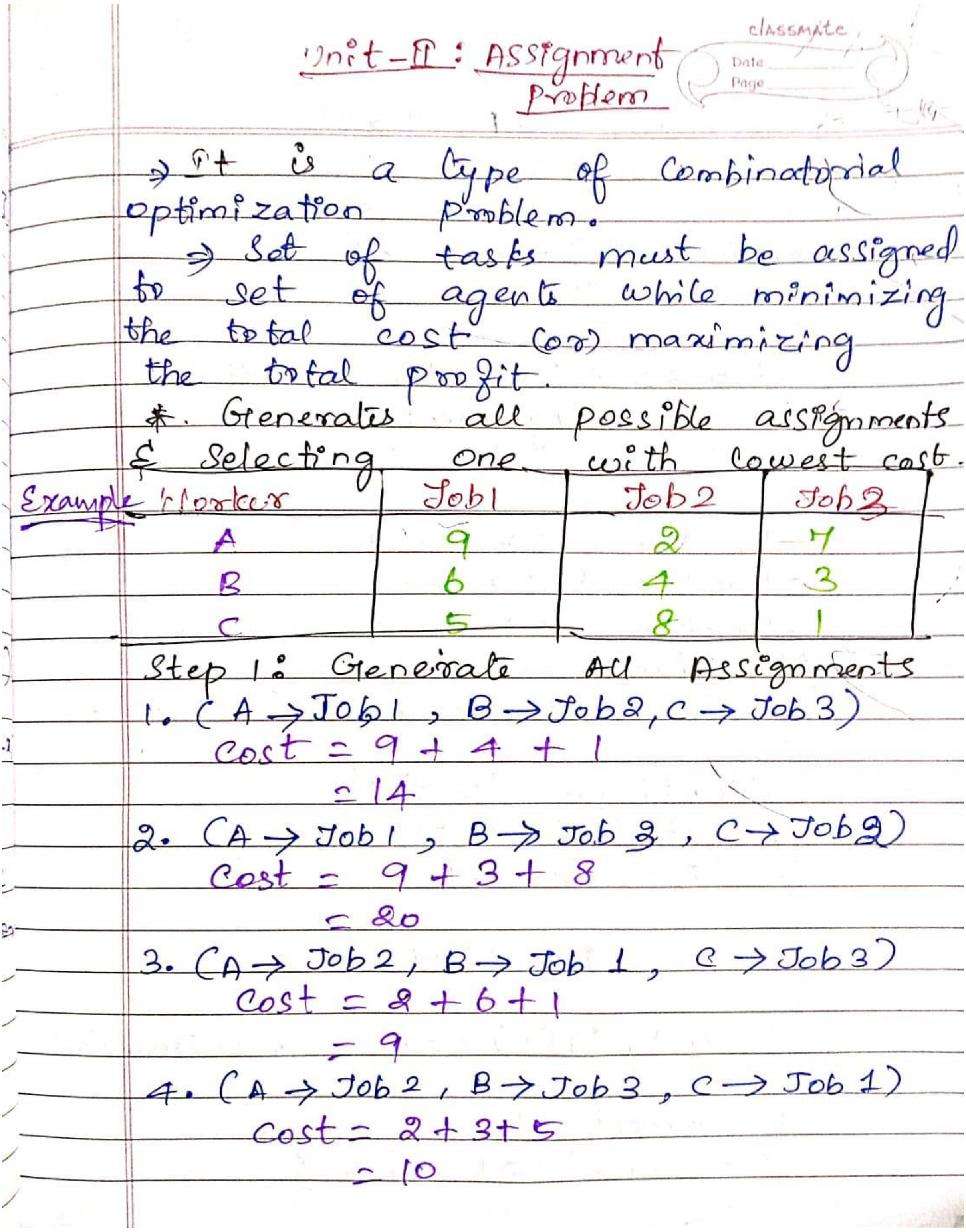
II YEAR/ IV SEMESTER

UNIT – II

BRUTE FORCE METHOD

Topic

Assignment Problem



Unit-II : Assignment Problem

→ It is a type of combinatorial optimization problem.

→ Set of tasks must be assigned to set of agents while minimizing the total cost (or) maximizing the total profit.

*. Generates all possible assignments & Selecting one with lowest cost.

Example

Worker	Job1	Job2	Job3
A	9	2	7
B	6	4	3
C	5	8	1

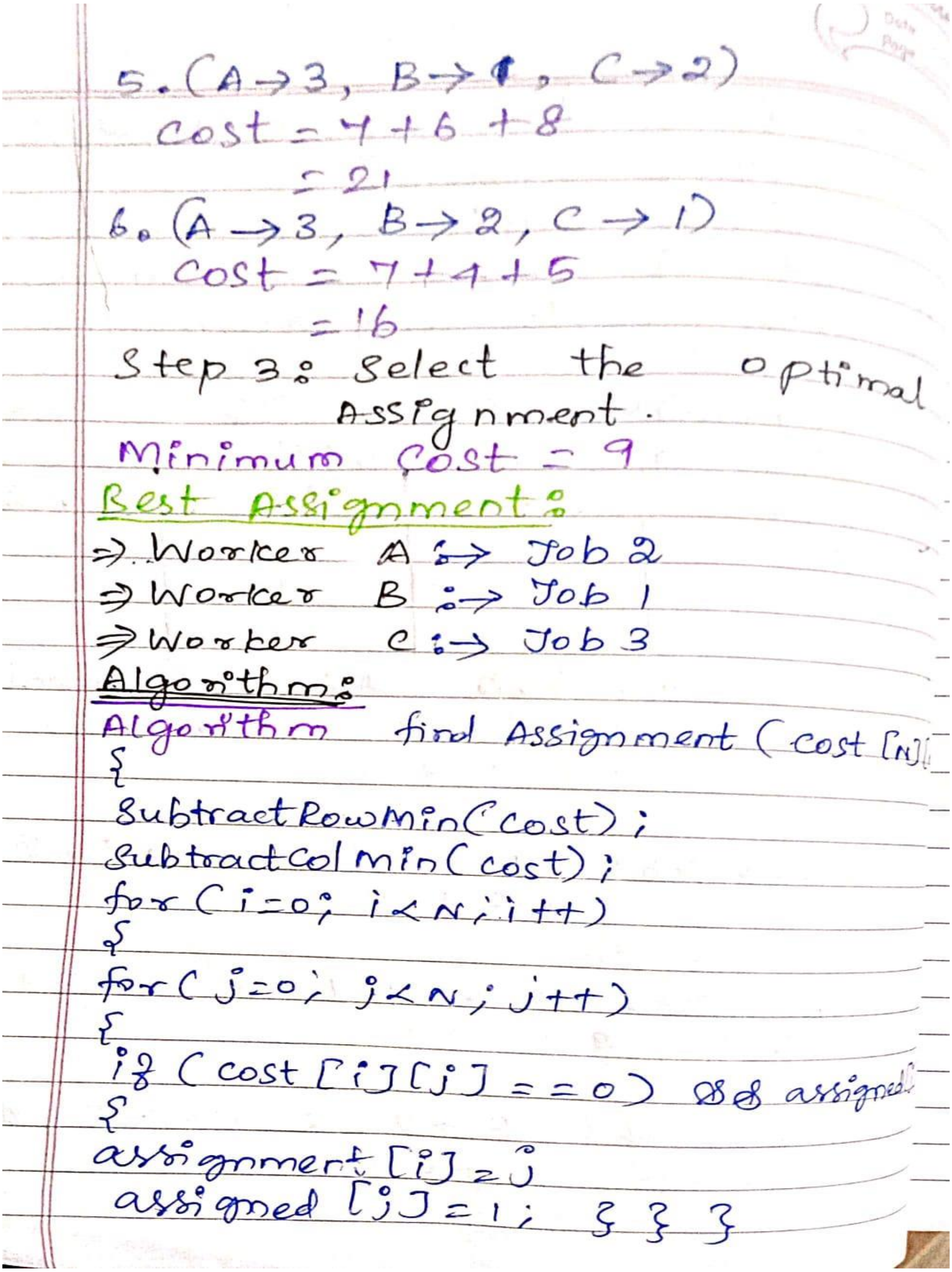
Step 1: Generate All Assignments

1. (A → Job1, B → Job2, C → Job3)
   Cost = 9 + 4 + 1
   = 14

2. (A → Job1, B → Job 3, C → Job2)
   Cost = 9 + 3 + 8
   = 20

3. (A → Job2, B → Job 1, C → Job3)
   Cost = 2 + 6 + 1
   = 9

4. (A → Job2, B → Job3, C → Job 1)
   Cost = 2 + 3 + 5
   = 10



5. (A → 3, B → 1, C → 2)
cost = 7 + 6 + 8
= 21

6. (A → 3, B → 2, C → 1)
cost = 7 + 4 + 5
= 16

Step 3: Select the optimal Assignment.

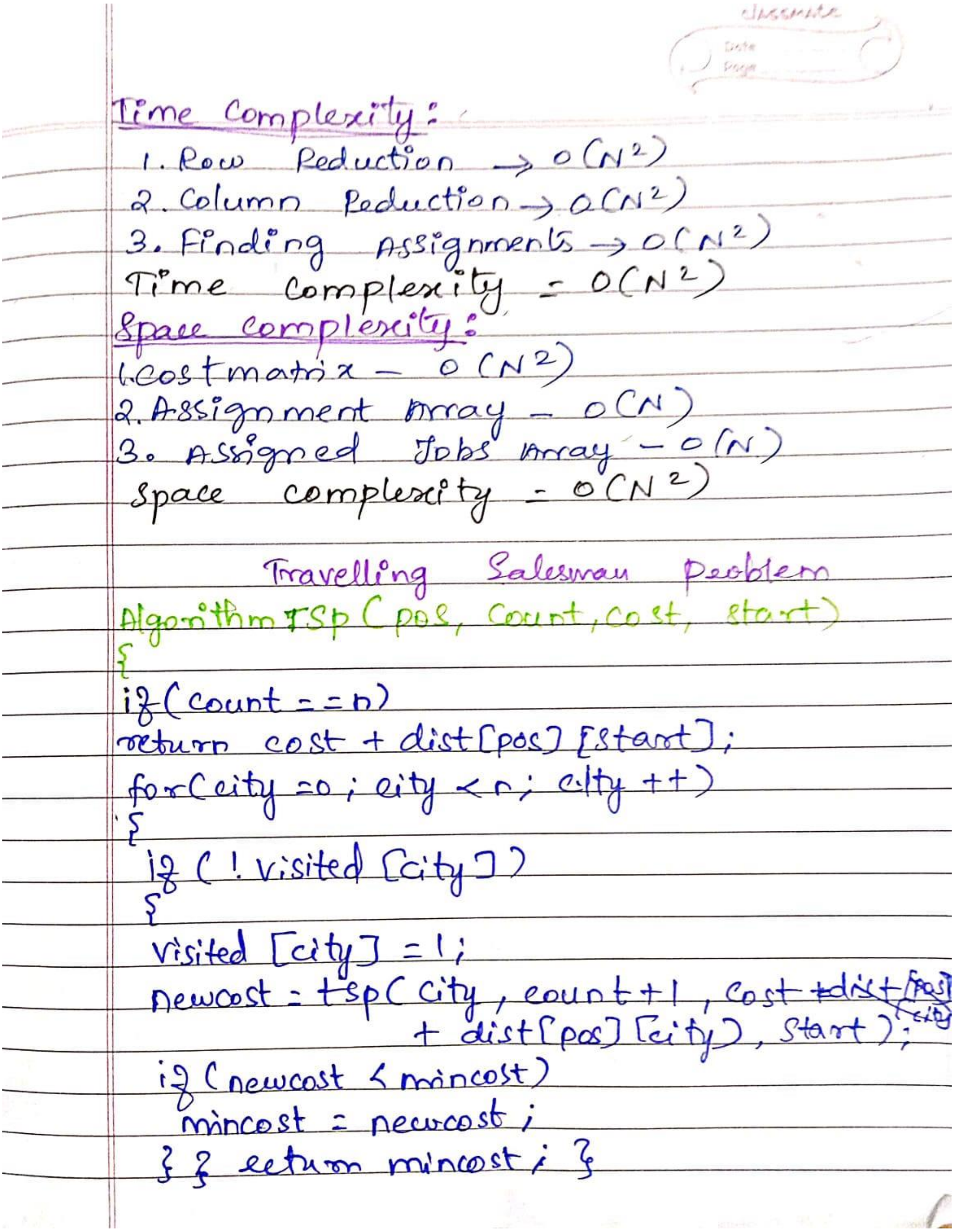
Minimum cost = 9

Best Assignment:

⇒ Worker A :→ Job 2
⇒ Worker B :→ Job 1
⇒ Worker C :→ Job 3

Algorithm:

```
Algorithm find Assignment (cost [N][
{
SubtractRowMin(cost);
SubtractColmin(cost);
for (i=0; i<N; i++)
{
for (j=0; j<N; j++)
{
if (cost [i][j] = = 0) && assigned[
{
assignment [i] = j
assigned [j] = 1; } } }
```



Time Complexity:

1. Row Reduction → O(N²)
2. Column Reduction → O(N²)
3. Finding Assignments → O(N²)

Time Complexity = O(N²)

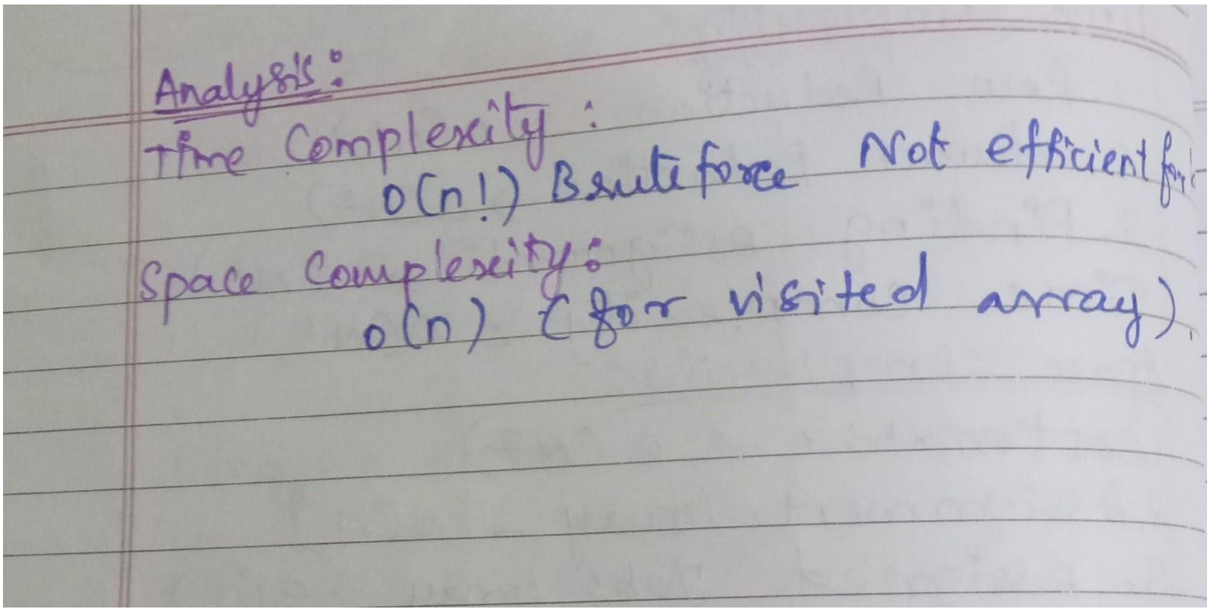
Space complexity:

1. costmatrix — O(N²)
2. Assignment Array — O(N)
3. Assigned Jobs Array — O(N)

Space complexity = O(N²)

Travelling Salesman problem

```
Algorithm TSP(pos, count, cost, start)
{
if(count == n)
return cost + dist[pos][start];
for(city = 0; city < n; city++)
{
 if(!visited[city])
 {
 visited[city] = 1;
 newcost = tsp(city, count+1, cost ~~+dist[pos][city]~~
               + dist[pos][city]), start);
 if(newcost < mincost)
  mincost = newcost;
 }} return mincost; }
```



Analysis:

Time Complexity:

O(n!) Brute force Not efficient for

Space Complexity:

O(n) (for visited array).