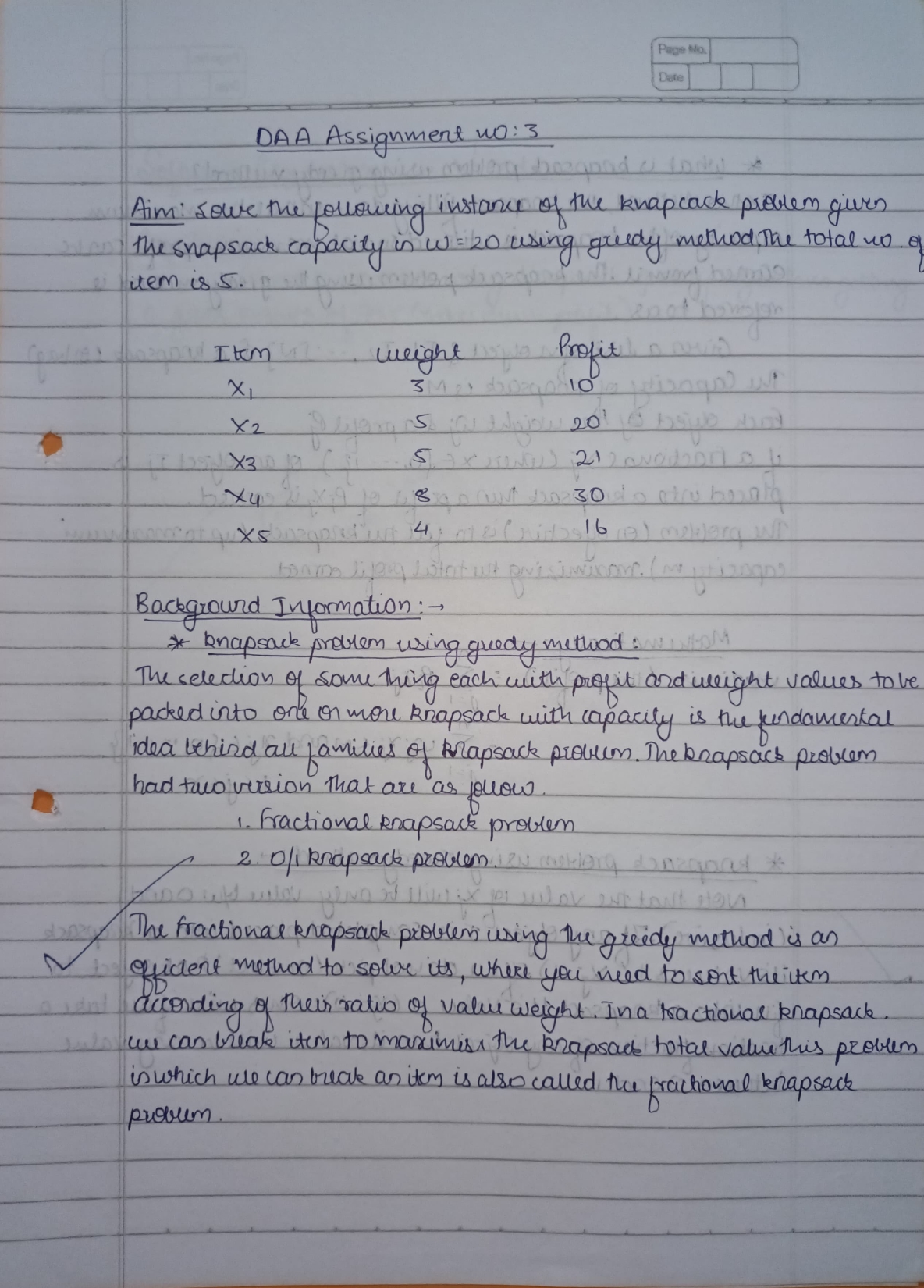
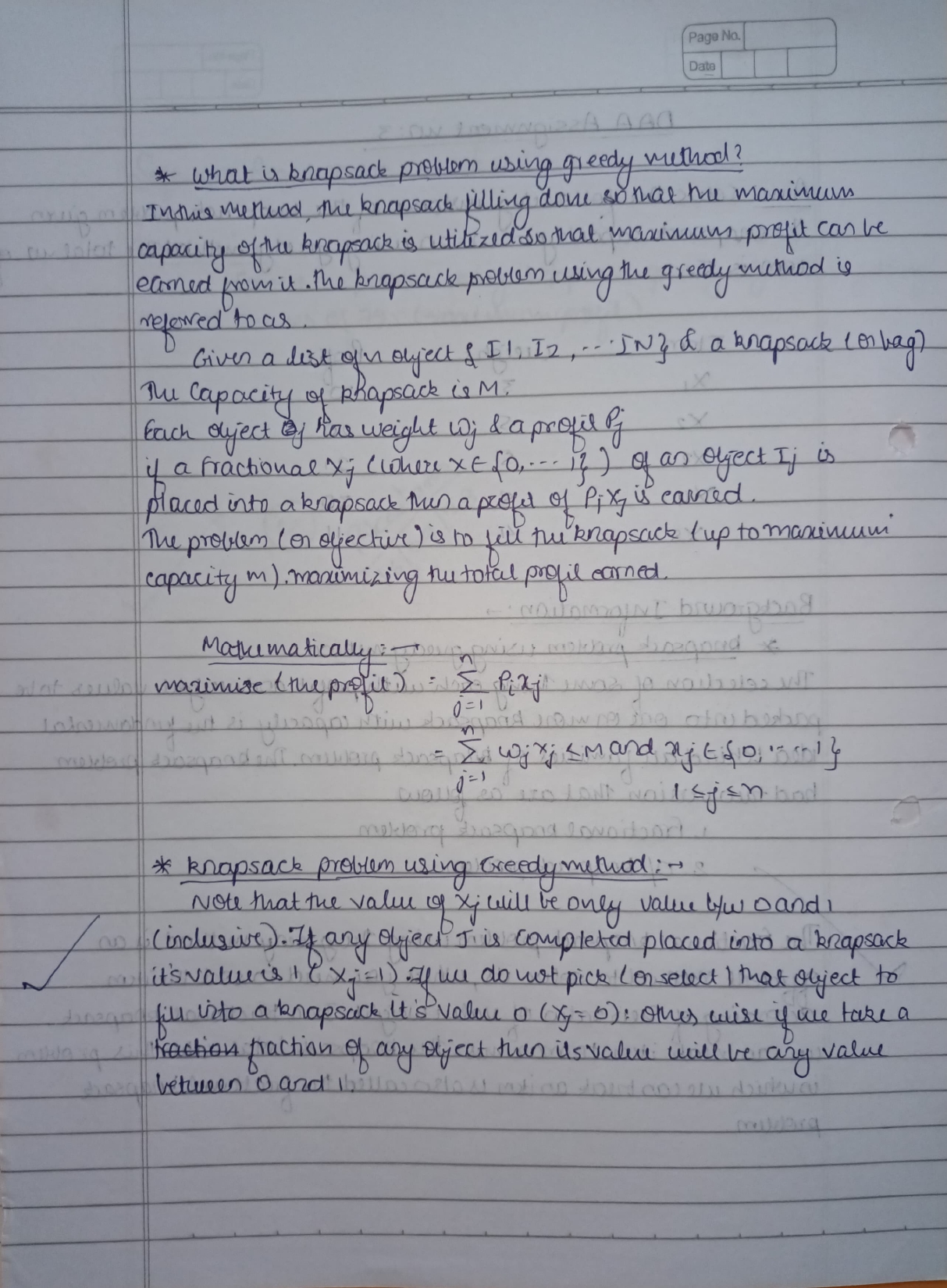
|  |  |  |  |
| --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Name: Siddhesh Dilip Khairnar | | | |
| Class: TY | Division: B | | Roll No: 372028 |
| Semester: V | | Academic Year: 2023-2024 | |
| Subject Name & Code: Design and Analysis of Algorithm: ADUA31202 | | | |
| Title of Assignment: Solve the following instance of the knapsack problem given the knapsack capacity in w=20 using greedy methods. The total number of items is 5. | | | |
| Date of Performance: 10-09-2023 | | Date of Submission: 16-09-2023 | |

**ASSIGNMENT NO. 3**





A piece of paper with writing on it

Description automatically generated

**Aim:** Solve the following instance of the knapsack problem given the knapsack capacity in w=20 using greedy methods. The total number of items is 5.

A table with numbers and a weight

Description automatically generated

**Problem Statement:** Use the knapsack greedy method to find maximum profit.

**Program Code:**

#include <iostream>

#include <algorithm>

using namespace std;

struct Item

{

    int value, weight;

    Item(int value, int weight)

    {

        this->value = value;

        this->weight = weight;

    }

};

bool cmp(struct Item a, struct Item b)

{

    double r1 = (double)a.value / (double)a.weight;

    double r2 = (double)b.value / (double)b.weight;

    return r1 > r2;

}

double fractionalKnapsack(int W, struct Item arr[], int N)

{

    sort(arr, arr + N, cmp);

    double finalvalue = 0.0;

    for (int i = 0; i < N; i++)

    {

        if (arr[i].weight <= W)

        {

            W -= arr[i].weight;

            finalvalue += arr[i].value;

        }

        else

        {

            finalvalue += arr[i].value \* ((double)W / (double)arr[i].weight);

            break;

        }

    }

    return finalvalue;

}

int main()

{

    int W = 20;

    Item arr[] = {{10, 3}, {20, 5}, {21, 5}, {30, 8}, {16, 4}};

    int N = sizeof(arr) / sizeof(arr[0]);

    cout << "Maximum value we can obtain = " << fractionalKnapsack(W, arr, N);

    return 0;

}

**Result:**

A black background with white text

Description automatically generated

**Conclusion:** Solved the problem using knapsack greedy algorithm, getting a final answer of 79.5 (maximum profit).