

CSE1002

Assignment 6

Knapsack Problems

1. 0 1 knapsack
2. fractional knapsack Problems

20BDS0405 (Bimal parajuli)

0 1 knapsack problem where division of material is not allowed.

```
C > Users > Bimal > Desktop > knapsack01.cpp > main()
//Using the recursion approach for 0 1 knapsack without breaking.
#include <iostream>
#include <algorithm>
using namespace std;

int bigger(int a, int b)
{
    if (a < b)
    {
        return b;
    }
    else
    {
        return a;
    }
}

int knapsack(int weight[], int value[], int n, int w)
{
    if (w == 0 || n == 0)
    {
        return 0;
    }
    if (weight[n - 1] > w)
    {
        return knapsack(weight, value, n - 1, w);
    }
    else
    {
        return bigger((value[n - 1] + knapsack(weight, value, n - 1, w - weight[n - 1])), knapsack(weight, value, n - 1, w));
    }
}

int main()
{
    int value[] = {60, 100, 120}; //array of profits
    int weight[] = {10, 20, 30}; //array of weights
    int w = 50; //max weight limit 50

    std::sort(value, value + 3); //optional sorting code but good to--
    std::sort(weight, weight + 3); //keep as value may be unsorted.

    int n = sizeof(value) / sizeof(value[0]);
    cout << "\nThe maximized profit is: " << knapsack(weight, value, n, w) << endl;

    return 0;
}
```

```
c:\Users\Bimal\Desktop>cd "c:\Users\Bimal\Desktop\" && g++ knapsack01.cpp -o knapsack01 && "c:\Users\Bimal\Desktop\"knapsack01

The maximized profit is: 220

c:\Users\Bimal\Desktop>
```

Ln 33, Col 2 Spaces: 4 LUT: 8 CRUE C++ Win32

Fractional knapsack problem where division of material is allowed:

```
C: > Users > Bimal > Desktop > fractional_knapsack.cpp >
#include <iostream>
#include <algorithm>
using namespace std;
typedef struct
{
    int v;
    int w;
    float d;
} Item;
void input(Item items[], int sizeOfItems)
{
    cout << "Enter total " << sizeOfItems << " item's values and weight" << endl;
    for (int i = 0; i < sizeOfItems; i++)
    {
        cout << "Enter " << i + 1 << " V ";
        cin >> items[i].v;
        cout << "Enter " << i + 1 << " W ";
        cin >> items[i].w;
    }
}
void display(Item items[], int sizeOfItems)
{
    int i;
    cout << "values: ";
    for (i = 0; i < sizeOfItems; i++)
    {
        cout << items[i].v << " ";
    }
    cout << endl;
    cout << "weight: ";
    for (i = 0; i < sizeOfItems; i++)
    {
        cout << items[i].w << " ";
    }
    cout << endl;
}
bool compare(Item i1, Item i2)
{
    return (i1.d > i2.d);
}
float knapsack(Item items[], int sizeOfItems, int W)
{
    int i, j, pos;
    Item mx, temp;
    float totalValue = 0, totalWeight = 0;
    for (i = 0; i < sizeOfItems; i++)
    {
        items[i].d = items[i].v / items[i].w;
    }
    sort(items, items + sizeOfItems, compare);
    for (i = 0; i < sizeOfItems; i++)
    {
        if (totalWeight + items[i].w <= W)
        {
            totalValue += items[i].v;
            totalWeight += items[i].w;
        }
        else
        {
            int wt = W - totalWeight;
            totalValue += (wt * items[i].d);
            totalWeight += wt;
            break;
        }
    }
    cout << "total weight in bag " << totalWeight << endl;
    return totalValue;
}
int main()
{
    int W;
    Item items[4];
    input(items, 4);
    cout << "Entered data \n";
    display(items, 4);
    cout << "Enter Knapsack weight \n";
    cin >> W;
    float mxVal = knapsack(items, 4, W);
    cout << "Max value for " << W << " weight is " << mxVal;
}
```

```
c:\Users\Bimal\Desktop>cd "c:\Users\Bimal\Desktop\  
& "c:\Users\Bimal\Desktop\"fractional_knapsack  
Enter total 3 item's values and weight  
Enter 1 V 60  
Enter 1 W10  
Enter 2 V 100  
Enter 2 W20  
Enter 3 V 120  
Enter 3 W30  
Entered data  
values: 60      100      120  
weight: 10      20       30  
Enter Knapsack weight  
50  
total weight in bag 50  
Max value for 50 weight is 240  
c:\Users\Bimal\Desktop>
```