Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Fall, Year:2021), B.Sc. in CSE (Day)

Course Title: Algorithm Lab
Course Code: CSE 206 Section: 201 DD

Lab Project Name: In Search of El'Dorado (Maximize your Profit)

Student Details

	Name	ID
1.	Md. Obaydullah Khan	201002093

Submission Date : January 3, 2022
Course Teacher's Name : Ms. Sultana Umme Habiba

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks: ………………………………… Signature: ....

Comments: .... Date: ....

Chapter 1

Introduction

1.1 Introduction

This project is a simple but real life necessary problem based project that will be used in every place where we have to choose some Items in range of limitation and those Items also have different value or cost or price, In that type of case everyone will try to get more profit from those Items or will try to get more valuable Items. In this case we can calculate maximum profit using 0-1 Knapsack algorithm but there is a problem, the algorithm doesn't say us which Items we have to collect for maximum profit. For solving this problem I made this project and this project will says us which Items we have to collect and also those weight and those values and finally it will says us our total profit.

1.2 Design Goals

There is an ancient myth about El'Dorado[1] (The mythical city of Gold). That is one day one man find an island when he was trying to find his way to home in deep forest of Amazon and this island is full of Gold, Silver , Diamond and others valuable things. There is a small boat with him that can carry one man and only some weight with him. For this reason the man can't take as many as valuable things with him and he thought he will come back here again and took more valuable things but next time he never find the island again. After that the man realize that if he took some others items instead of those item he took , he will get more money from selling those items and he can invest more money for searching that Island called El'Dorado (The city of GOLD). After knowing about this myth I thought about 0-1 knapsack algorithm that can find maximum profit from that type of scenario. The 0-1 knapsack algorithm can only find maximum profit count but cant shows which Items we have to take for maximum profit. For taking optimum items I have to calculate manually and sometimes that's very boring. So, I decided to make something that will shows us which Items we have to take and their all property like their value and their weight. And this will shows us The Optimum Profit from this Items.

2nd Edition: I have added a loading screen with this program. And add some feature for those people who carry a stone cutter with him and he can take any items part for fulfil of his knapsack.

Its mean I implement fractional knapsack beside zero one knapsack algorithm .

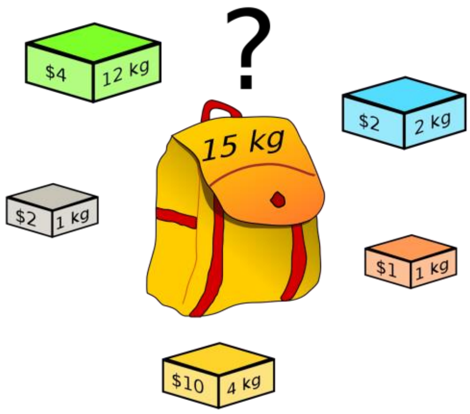
Chapter 2

Implementation of the Project

2.1 The 0 – 1 Knapsack Algorithm

The knapsack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. The problem often arises in resource allocation where the decision makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively.

The knapsack problem has been studied for more than a century, with early works dating as far back as 1897.[1] The name "knapsack problem" dates back to the early works of the mathematician Tobias Dantzig (1884–1956),[2] and refers to the commonplace problem of packing the most valuable or useful items without overloading the luggage.



2.1.1 Pseudocode

```
Input: Weights, Values
Output: P[n, W]

for w = 0 to W do
   P[0, w] = 0
end
for i = 1 to n do
  P[i, 0] = 0
  for w = 1 to W do
     if wi ≤ w then
          P[i, w] = max{vi + P[i - 1, w - wi], P[i - 1, w]}
     end
     else
          P[i, w] = P[i - 1, w]
     end
   end
end
```

2.2 Code

```
int ZOKS(int W, int wt[], int val[], int n) /// 0-1 Knapsack Algorithm
      & Details of Taken Items Function

{

      int i, w;

      int K[n + 1][W + 1];


      for (i = 0; i <= n; i++) /// Build table K[][] in bottom up
      manner

    {

       for (w = 0; w <= W; w++)

       {

       if (i == 0 || w == 0)

           {

               K[i][w] = 0;

           }

       else if (wt[i - 1] <= w)

           {

               K[i][w] = max(val[i-1] + K[i-1][w-wt[i-1]], K[i-
      1][w]);

           }

       else

             K[i][w] = K[i - 1][w];

      }

      }


      int res = K[n][W];/// Storing result for finding Items

      int profit = res; /// Storing profit for returning profit to the
      main function


      w = W;
```

```
        cout << "  Optimum Profitable Items Detail list :\n" << endl <<
        endl;

        cout << "\tSerial Number    \tWeight \t\t Profit\n";

    cout << "\t_______________________________________________\n\n";


        for (i = n; i > 0 && res > 0; i--)

          {

              if (res == K[i - 1][w]) /// Item NOT Taken

                  continue;

              else /// This Item IS Taken

              {

                  cout << "\t\t"<<i << "\t\t  "<<wt[i - 1] << "\t\t  "
        << val[i-1] << endl;

                  cout << endl;

                  res = res - val[i-1];

                  w = w - wt[i-1];

              }

          }


        return profit; /// Returning profit To The Main Function

}


int FKS(int W, int wt[], int val[], int n) /// Fractional Knapsack
        algorithm & Details of Taken Items function

{

    vector <pair<int,int>> a(n);

    for(int i=0 ; i<n ; i++)

    {

        a[i].first = val[i];

        a[i].second = wt[i];

    }


    sort(a.begin(), a.end() , comp);


    int ans = 0 ;
```

```
        cout << "  Optimum Profitable Items Detail list :\n" << endl <<
          endl;

          cout << "\tWeight    \tFree Space After Taking \t\t Profit\n";

        cout <<
          "\t____________________________________________________________
          __\n\n";


        for(int i = 0 ; i<n ; i++)

        {

            if(W >= a[i].second)

            {


                ans += a[i].first;

                W -= a[i].second;

                cout << "\t" << a[i].second  <<"    \t\t\t" << W << "
          " << a[i].first << endl;

                continue;

            }

            double vW = (double) a[i].first/a[i].second;

            int x = W*vW;

            ans += vW*W;

            W = 0 ;

            cout << "\t" << a[i].second  <<"    \t\t\t"<< W << "
          "  << x<< endl;

            break;


        }

        cout << endl;

        return ans;

}
```

Full code link : https://ideone.com/VpT9Q2

Chapter 3

Performance Evaluation

3 Results

```
####################################
##================================##
##=     IN SEARCH OF EL'DORADO     =##
##================================##
##================================##
##=     Maximize Your Profit!!     =##
##================================##
####################################


        LOADING.......!!!
```

Img 1 : Loading Screen

```
              Have You any Stone cutter With YOu???


    ##-----------------------------------------------------------##
    ##-----------------------------------------------------------##
    ##-     1.      I Have Stone Cutter With Me!!               -##
    ##-----------------------------------------------------------##
    ##-----------------------------------------------------------##
    ##-     2.      I Don't Have Any Stone Cutter With me!!     -##
    ##-----------------------------------------------------------##
    ##-----------------------------------------------------------##
```

Img 2 : 2nd screen (0-1 or fractional knapsack choosing)

```
Optimum Profitable Items Detail list :


      Weight          Free Space After Taking                 Profit
      ______________________________________________________________

      40                      210                              643
      26                      184                              400
      15                      169                              132
      30                      139                              92
      63                      76                              100
      88                      0                                 42


If You Take All Items of The list, Your Profit Will Be 1409.....!!!
```

Img 2 : List for fractional Knapsack

```
Total Items Count(s) is 7

      Items Weight & Value List :
      __________________________

              15       132

              40       10

              26       400

              63       100

              88       49

              40       643

              30       92


Optimum Profitable Items Detail list :

      Serial Number           Weight           Profit
      _______________________________________________

              7                 30               92

              6                 40               643

              4                 63               100

              3                 26               400

              2                 40               10

              1                 15               132

If You Take All Items of The list, Your Profit Will Be 1377.....!!!
```

Img 4 : 0-1 Knapsack Screen

Chapter 4

Conclusion

4.1 Discussion

This is an algorithm base simple unique real life problem project. We can easily find optimum items from any list for any limitation. As though this project based on a mythical city called EL'DORADO but the project can be use in real life . Mainly this projects Loading screen is nothing but a simple decoration. After finishing loading screen the main algorithmic page will be view. This page will shows us how many Items in total we have an the list of their weight and value , basically this part is our input part , The output parts shows us optimum items details table that shows us items serial number those we have to take for maximum profit, the table also shows us their weight and their values . And finally shows us optimum profit from those items.

4.2 Practical Implications

We can implement in this project for collecting much valuable things from unfamiliar place like underground mine or under water ocean. We cant carry unlimited items at a time , so , if we use this project for mining under water and under ground then we can find out easily that which items we will carry with us and that will be more profitable.

References

[1] *Eldorado*, by Neil Young (1989)

[2] Kellerer, Hans; Pferschy, Ulrich; Pisinger, David (2004). Knapsack Problems. Springer. doi:10.1007/978-3-540-24777-7. ISBN 978-3-540-40286-2. MR 2161720.