14. Problem Statement: Container Loading

**Problem Analysis:**

In this algorithm, different container may have different weights. Let wi be the weight of the ith container, 1 ≤ i ≤ n. The cargo capacity of the ship is c. We wish to load the ship with the maximum number of containers.

This problem can be formulated as an optimization problem in the following way: Let xi be a variable whose value can be either 0 or 1. If we set xi to 0, then container i is not to be loaded. If xi is 1, then the container is to be loaded. We wish to assign values to the xi s that satisfy the constraints. Every set of xis that satisfies the constraints is a feasible solution. Every feasible solution that maximizes xi is an optimal solution.

**Algorithm:**

Algorithm ContainerLoading (c, capacity, numberOfContainers, x)

*// Greedy algorithm for container loading.*

*// Set x [i] = 1 iff container c [i], i≥ 1 is loaded.*

{

*// sort into increasing order of weight*

Sort (c, numberOfContainers);

n := numberOfContainers;

*// Initialize x*

for i := 1 to n do

x [i] := 0;

*//select containers in order of weight*

i := 1;

while (i ≤ n and c [i].weight ≤ capacity)

{

*//enough capacity for container c [i].id*

x [c[i].id] := 1;

capacity -= c [i].weight; *// remaining capacity*

i++;

}

}

**Source Code:**

#include<bits/stdc++.h>

using namespace std;

struct Data

{

int id, weight;

bool operator<(Data A)

{

if(weight<A.weight)

return true;

return false;

}

};

void containerLoading(Data c[], int capacity, int numberOfContainers, int x[])

{

int n = numberOfContainers;

sort(c,c+n);

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

x[i] = 0;

for(int i = 0; i < n && c[i].weight<=capacity; i++)

{

x[c[i].id] = 1;

capacity -= c[i].weight;

}

}

int main()

{

int n, capacity;

cout<<"Please enter num of containers and capacity:\n";

cin>> n >> capacity;

int ar[n];

Data c[n];

int x[n];

cout<<"Enter the container size:\n";

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

cin>> ar[i];

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

{

c[i].id = i+1;

c[i].weight = ar[i];

}

cout<<endl;

containerLoading(c,capacity,n,x);

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

{

if(x[c[i].id]==1)

cout<<c[i].id<<' '<<c[i].weight<<'\n';

}

return 0;

}

**Sample Input:**

Please enter num of containers and capacity:

8 400

Enter the container size:

100 200 50 90 150 50 20 80

**Sample Output:**

7 20

3 50

6 50

8 80

4 90

1 100