**dSet.h**

#ifndef DSET\_H

#define DSET\_H

#include "dUSet.h"

#include <string>

class dSet: public dUSet{

public:

dSet();

dSet(int a[], int sz);

void insert(int target);

};

#endif

**dSet.cpp**

#include "dSet.h"

dSet::dSet(): dUSet(){}

dSet::dSet(int a[], int sz): dUSet(){

int tmp;

//erase duplicates

for(int i = 0; i < sz-1; i++){

for(int j = i+1; j < sz; j++){

if(a[i] == a[j]){

sz--;

for(int k = i; k < sz; k++){

a[k] = a[k+1];

}

j--;

i--;

}

}

}

//Fill array in ascending order

for(int i = 0; i < sz-1; i++){

for(int j = i+1; j < sz; j++){

if(a[i] > a[j]){

tmp = a[i];

a[i] = a[j];

a[j] = tmp;

j--;

}

}

}

//using InsertAt to fill dSet

for(size\_type i = 0; i < sz; i++){

dynamicBag::insertAt(a[i], i);

}

}

void dSet::insert(int target){

size\_type moveCounter = 0;

if(count(target) == 0){

for(int i = 0; i < target; ++i){

if(count(i) == 1){

moveCounter++;

}

}

dynamicBag::insertAt(target, moveCounter);

}

}

**main.cpp | Only for Problem 2(dSet)**

#include "dUSet.h"

#include "dSet.h"

#include <iostream>

using namespace std;

int main(){

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Testing Problem 3, Constructors"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

dSet baseDSet = dSet();

int unsortedArray[12] = {4,2,3,1,5,1,1,2,3,4,2,2};

cout<<"Array we start with: ";

for(int i = 0; i < 12; i++){

cout<<unsortedArray[i] << ", ";

}

cout<<""<<endl;

dSet sortedDSet = dSet(unsortedArray, 12);

cout<<"Sorted set: " << sortedDSet<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Testing Problem 3, Insert Function"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Inserting 10 into the set." <<endl;

sortedDSet.insert(10);

cout<<"Set after inserting 10 : " << sortedDSet<<endl;

cout<<"Inserting 2 into the set." <<endl;

sortedDSet.insert(2);

cout<<"Set after inserting 2 : " << sortedDSet<<endl;

cout<<"Inserting 8 into the set." <<endl;

sortedDSet.insert(8);

cout<<"Set after inserting 8 : " << sortedDSet<<endl;

cout<<"Inserting 200 into the set." <<endl;

sortedDSet.insert(200);

cout<<"Set after inserting 200 : " << sortedDSet<<endl;

cout<<"Inserting 16 into the set." <<endl;

sortedDSet.insert(16);

cout<<"Set after inserting 16 : " << sortedDSet<<endl;

cout<<"Inserting 14 into the set." <<endl;

sortedDSet.insert(14);

cout<<"Set after inserting 14 : " << sortedDSet<<endl;

cout<<"Inserting 12 into the set." <<endl;

sortedDSet.insert(12);

cout<<"Set after inserting 12 : " << sortedDSet<<endl;

return(0);

}

**Result from running main**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Testing Problem 3, Constructors

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Array we start with: 4, 2, 3, 1, 5, 1, 1, 2, 3, 4, 2, 2,

Sorted set: 1 2 3 4 5

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Testing Problem 3, Insert Function

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Inserting 10 into the set.

Set after inserting 10 : 1 2 3 4 5 10

Inserting 2 into the set.

Set after inserting 2 : 1 2 3 4 5 10

Inserting 8 into the set.

Set after inserting 8 : 1 2 3 4 5 8 10

Inserting 200 into the set.

Set after inserting 200 : 1 2 3 4 5 8 10 200

Inserting 16 into the set.

Set after inserting 16 : 1 2 3 4 5 8 10 16 200

Inserting 14 into the set.

Set after inserting 14 : 1 2 3 4 5 8 10 14 16 200

Inserting 12 into the set.

Set after inserting 12 : 1 2 3 4 5 8 10 12 14 16 200