Data Structures
CS 246 - 040
Department of Physics and Computer Science
Medgar Evers College
Exam 2

Instructions:

- The exam requires writing a complete header file and cpp file within an hour and 50 minutes. It requires completing tasks in three sections.

- Accompanying this file is a template cpp file and header files. You must modify the cpp file; however, you cannot add additional libraries to or remove any libraries from the file. Furthermore, you cannot create any additional classes and/or structs. All other modifications are allowed.

- Tables can be constructed using a spreadsheet application instead of in the the cpp file.

- Your submissions must be submitted to the Exams directory of your github repository and/or as attachments on Google classroom under the Exam02 assessment. The files must have the accurate extensions.

- Cheating of any kind is prohibited and will not be tolerated.

- Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the exam.

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, AT THE BEGINNING OF YOUR SUBMISSION(S), ADD A COMMENT THAT CONSISTS OF YOUR NAME AND THE DATE

Grading:

Section	Maximum Points	Points Earned
Runtime	5	
Implementation	5	
Problem Solving	10	
Total	20	

Runtime

1. Construct the runtime table and determine the runtime function of the following function for the worst-case scenario. Let the cost of every operation be 1. Write the function in terms of *n* where *n* represents the int parameter which is the size of the array parameter.

```
void F(string data[],int n)
{
  for(int i = 0;i < n;i += 1)
  {
    data[i] = "[";

    for(char j = '0';j <= '9';j += 1)
    {
      data[i] += j;
    }
    data[i] = "]";
  }
}
```

Implementation

2. A bag data structure is a collection of items. A *Bag* container class of a bag data structure contains the fields

```
template<class T>
class Bag
{
  private:
  Array<Item<T>> data;
  int size;
};
```

where *Item<T>* is a struct of the form

```
template<class T>
struct Item
{
  T value;
  int count;
};
```

and *size* represents a number of distinct items in the bag data structure.
Given that the items of the bag is stored in ascending order in *data*, write the definition of the insertion method of *Bag* whose header is

```
void Insert(const T& item)
```

If *item* already exists in the bag [the *value* member of one of elements of *data* is equal to *item*], it increments the *count* member of that element by 1. Otherwise, if *size* is less than the capacity of *data*, it adds an *Item* object whose *value* is assigned *item* and *count* is assigned 1 to *data*, and it increments *size* by 1.

Problem Solving

3. Write the definition of the void function named `Sort()` whose header is

```
template<typename T>
void Sort(dn::Node<T>* data)
```

It sorts the linked list referenced by *data* in ascending order using the selection sort algorithm.

4. Write the definition of the bool function named `IsValid()` whose header is

```
bool IsValid(string str)
```

It returns true if *str* is empty or represents a valid enclosure of parentheses, `()`, and square braces, `[]`. For instance, the callers `IsValid("[([])]")` and `IsValid("[(])")` will return `true` and `false` respectively.