



Data Structures

CS 246

Department of Physics and Computer Science

Medgar Evers College

Exam 1

Direction: Modify the "exam01.cpp" file in your Exams directory of your GitHub repository; and then, submit your modified work in the Exams directory of your GitHub repository or Dropbox, or in your Exam01 google classroom assignment. You can only use the libraries included in the accompanying header file and the cpp file.

Problem	Maximum Points	Points Earned
1	5	
2	5	
3	5	
4	5	
Total	20	

Problems

1. Write the definition of the function `MinimumDistance()` whose header is

```
int MinimumDistance(Array<int>& data)
```

It returns the minimum distance between adjacent elements of *data*. For instances, if *data* = [1,4,6,5,9], it will return 1 due to the adjacent elements (6,5). It is important to note that distance is never negative. Furthermore, it returns 0 if *data* has a size of 1 or is empty.

2. Write the definition of the function `Similar()` whose header is

```
template <typename T>
bool Similar(Array<T>& ar1, Array<T>& ar2)
```

It returns true if the elements of *ar1* and *ar2* with the same index have the same value, but *ar1* and *ar2* are not necessarily the same length; otherwise, it returns false.

3. Write the definition of the function `MidAppend()` whose header is

```
template <typename T>
void MidAppend(Array<T>& data, Array<T>& addon)
```

It appends the content of *addon* to *data* starting after the middle element of *data*. It is important to note that you may have to resize *data* in order to hold both the original values from *data* and the values from *addon*. For instances, if *data* = [a, b, c, d, e] and *addon* = [f, g, h, i, j]; then after the call of the function, *data* = [a, b, f, g, h, i, j, c, d, e].

4. Construct the runtime table and calculate the worst-case scenario runtime for

```
int P(int a[],int low,int high)
{
    int t;
    int lo = low;

    if(lo > high)
    {
        lo = high;
    }
    int hi = (high + low) - lo;
    int i = lo - 1;
    int pivot = a[hi];

    for(int j = lo;j < hi;j += 1)
    {
        if(a[j] < pivot)
        {
            i += 1;
            t = a[i];
            a[i] = a[j];
            a[j] = t;
        }
    }
    t = a[i+1];
    a[i+1] = a[hi];
    a[hi] = t;
    return (i + 1);
}
```

where *high* and *low* are valid indices of *a*. Likewise, let *n* be the length between *high* and *low* inclusively. Furthermore, assume the operation time cost is 1 for every operation. The table must be a comment.