CPSC 2150 – Algorithms and Data Structures II
Lab2: Complexity Analysis
Total - 50 Marks

Learning Outcomes

- Design algorithm with a given time complexity
- Calculate the time/space complexity

Resources

- Chapter 4 of the text book

Description

1. **[8 marks]** Determine the best and worst complexity (time and space) of the following function in terms of O and Ω **(answers.pdf)**.

```
int noName(int n, int target, int limit){
    int* ptr = nullptr;
    for (int i=0; i<n ; i++){
        ptr = new int(rand()%limit);
        if ( *ptr == target )
            return i;
    }
    return -1;
}
```

2. **[6 marks]** Find the computational time complexity for the following loops: **(answers.pdf)**

a.
```
for (cnt4 = 0, i = 1; i <= n; i *= 2)
    for (j = 1; j <= i; j++)
        cnt4++;
```

b.
```
for (cnt2 = 0, i = 1; i <= n; i++)
    for (j = 1; j <= i; j++)
        cnt2++;
```

c.
```
for (cnt3 = 0, i = 1; i <= n; i *= 2)
    for (j = 1; j <= n; j++)
        cnt3++;
```

3. **[15 marks]** Comparison of running times. This question is to give one an idea of the rate at which functions grow. For each function f(n) and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes f(n) microseconds. Note that logn means the logarithm in base 2 of n. Some entries have already been filled to get you started. For simplicity 2^{10} is considered 1000. **(answers.pdf)**

	1 Second	1 Hour	1 Month	1 Century
$\log n$	$\approx 10^{300000}$			
$\sqrt{n}$				
n				
$n \log n$				
n^2				
n^3				
2^n				
$n!$		12		

4. In an array of n keys, each entry may have one of the values; red, white or blue. **(answers.pdf).**
   - **A. [15 marks]** Give a linear time algorithm for rearranging the keys so that all the reds come before all the blues and all the blues come before all the whites. The only operations permitted on the keys are:
     - Examination of a key to find out what color it is.
     - A swap (interchange of positions) of two keys specified by their indices.
   - **B. [5 bonus marks]** design your algorithm with the best space complexity.
   - **C. [6 marks]** Implement your algorithm in C++. Use dynamic arrays. (rearrange.cpp)

Submit to D2L

Submit a file named **StudentNumber-lab2.zip** by the end of the lab time. For example, if your student number is 10023449, the submitted file must be named as **10023449-lab2.zip**.