COEN 346 Programming Assignment #1

By:Niroigen Suntharam (ID: 27890540)

# Methods

## int FindPivot(int left, int right);

### Description

This method will just return the pivot point between left and right value. For example, if the values given were 0 for left and 10 for right, then this method would return 5.

### Pseudocode

FindPivot(int left, int right) {  
 return ((right + 1) + left) / 2;  
}

## bool IsDefectiveSubarray(const int \* arr, int start, int end);

### Description

Will check whether the given subarray contains at least one defective light bulb or not.

### Pseudocode

IsDefectiveSubarray(int[] array, int start, int end)  
 FOR i = 0 to end  
 IF array[i] == 0

RETURN TRUE  
  
 RETURN FALSE

## void FindDefective(int \* arr, int left, int right);

### Description

This method is the main method that will be used to find the defective light bulbs. It will be called by multiple threads. It will keep track of the number of threads that are running. This method will find the pivot point of the array and check whether the left half of the array or the right has defective light bulb or not. This part is important, since a thread will only be created if there is a defective light bulb on either side of the pivot. If there is a defective light bulb, a thread would be created and calling the FindDefective method. This recursive function’s base case would be when both the left and right values are equal to each other, and see whether the value looking at is a defective light bulb or not.

### Pseudocode

FindDefective(int[] array, int left, int right)  
 SET num\_threads++;  
  
  
 IF left == right  
 IF array[left] == 0  
 print “Defective light bulb at “ + left  
 RETURN  
   
   
  
 IF NOT IsDefectiveSubarray(arr, left, right)

RETURN  
  
 pivot = FindPivot(left, right)   
  
 RUN THREAD left\_thread(FindDefective, arr, left, pivot - 1);  
 RUN THREAD right\_thread(FindDefective, arr, pivot, right);

## int ReadFile(const std::string& file);

### Description

This method is used to extract the contents of the file it is trying to read from. The first line will always contain the number of light bulbs there are in the file. Which will then be used to initialize the array containing the light bulbs to the size provided. Next it will extract every other line and save it.

### Pseudocode

int ReadFile(string fileName)  
 LINE  
  
 myfile(fileName)  
  
 numberOfLightBulbs = 0  
  
 isFirstLine = FALSE  
  
 IF CAN OPEN myfile  
 line\_num = 0;  
 WHILE (EXTRACT CURRENT LINE)  
 IF isFirstLine  
 numberOfLightBulbs <- EXTRACT LINE  
 CREATE lightBulbs[]  
 isFirstLine = FALSE  
 ELSE   
 light\_bulbs[line\_num++] = std::stoi(line);  
 CLOSE myfile  
  
 ELSE OUTPUT UNABLE\_TO\_OPEN\_FILE;  
  
 RETURN numberOfLightBulbs