Abhishikth Daniel Merugu (001548340)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 3**

* **Task**

1. Implemented height weighted quick union with Path compression in UF\_HWQUPC class
2. Created UnionFindClient to take predefined number of sites and find the number of connections made
3. Deriving the relation between sites and connection count

* **Relationship Conclusion**

M = (N \* log(N))/2

Here, M = Connection count and N = number of sites

1. **Output**

**Text

Description automatically generated**

1. **Graphical Representation**

**Chart, line chart

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| **Sites** | **Count** | **(N log N)/2** |
| **1000** | **4165** | **4982.892** |
| **2000** | **8186** | **10965.78** |
| **4000** | **15094** | **23931.57** |
| **8000** | **34353** | **51863.14** |
| **16000** | **67324** | **111726.3** |
| **32000** | **181775** | **239452.5** |
| **64000** | **368353** | **510905.1** |

**Chart, line chart

Description automatically generated**

From the above two graphs and tabulated result we can infer that graphs between Sites vs Count and Sites vs (N log N)/2 are equal.

Hence the relation is M = (N \* log(N))/2

Here, M = Connection count and N = number of sites

* **Unit tests result:**

**A screenshot of a computer

Description automatically generated with medium confidence**

Code:

package edu.neu.coe.info6205.union\_find;  
  
import java.util.Random;  
  
public class UnionFindClient {  
 static int *cCount* =0;  
 public static int RandomCount(int n) {  
 Random random = new Random();  
 return random.nextInt(n);  
 }  
  
 public static void main(String[] args) {  
 int[] exp= {1000,2000,4000,8000,16000,32000, 64000};  
 for(int i=0;i<exp.length;i++)  
 {  
 int n = exp[i];  
  
 *cCount* =0;  
 UF\_HWQUPC uf\_hwqupc = new UF\_HWQUPC(n);  
 while(uf\_hwqupc.components() != 1)  
 {  
  
 int p= *RandomCount*(n);  
 int q= *RandomCount*(n);  
 *count*();  
 if(p==q)  
 continue;  
 if(uf\_hwqupc.connected(p, q)) continue;  
 else {  
 uf\_hwqupc.union(p, q);  
  
  
 }  
  
 }  
 System.*out*.println("Sites= " + n +" Connection Count= "+ *cCount*);  
 }  
 }  
 public static int count()  
 {  
 return *cCount*++;  
 }  
  
}