**ALGORITHMS**

**Project 2**

**Union Find Algorithm**

**By**

**Jaswanth Datt Meka (jm212)**

**Professor**

**Dr. Zhong-Hui Duan**

**Academic Integrity Pledge**

Program: Message Encryption using RSA Algorithm

Programmer: - Jaswanth Datt Meka

Academic Integrity Pledge

* All source code and documentation used in my program is either my original work, or was derived, by me, from the source code published in the text book for this course or presented in class. Any source code in this project that is derived from code published elsewhere is documented as such so the original author receives due credit.
* I have not used source code obtained from another student, or any other unauthorized source, either modified or unmodified. I have not helped another student write their program by providing a printed or electronic copy of my solution.
* I have not discussed coding details about this project with anyone other than my instructor. I understand that I may discuss the concepts of this program with other students, and that another student may help me debug my program. However, the responsibility to write each program belongs solely to the program's author.
* I have violated neither the spirit nor letter of these restrictions.

**Signature of Programmer**  Date: 10/22/2014

Jaswanth Datt Meka **INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Introduction | |  | 4 |  |
| 2 | Algorithm- Project Description | | | 4 |  |
| 3 | Conclusion |  |  | 7 |  |
| 4 | References | |  | 7 |  |

**Union Find Algorithm**

1. **Introduction**

Union Find Algorithm

Union Find is an algorithm which uses a [disjoint-set data structure](http://www.algorithmist.com/index.php?title=Disjoint-set_data_structure&action=edit&redlink=1) to find the number of clusters. It has many applications in Social Networking (interactions), Electricity flow (conduction) ,Water Flow(porous) etc.

Union is used to merge two smaller subsets and form a bigger subset. For efficiency it is always suggestible to merge smaller one’s into larger one’s.

Find is used to search for the element for its root.

**Algorithm:**

* union( A, B ) - merge A's set with B's set
* find( A ) - finds what set A belongs to

**Union**:

func union( var setA, var setB )

var *rootA* = find( setA ), *rootB* = find( setB )

if ( *rootA* is equal to *rootB* )

return

else

set *rootB* as *rootA's* parent

end func

**Find:**

func find( var element )

while ( element is not the root )

element = element's parent

return element

end func

**Complexity Analysis : Union Find**

**Implementation Difficulties:**

* To write an  *isConnected()* Method ,which finds if an element is connected or not according to certain priority of Top/Right/Bottom/Left
* Implementation of find( ) to check all connected elements of a same root.
* Finding out if maze can be percolated by checking connectivity between top row & last row elements.

1. **Observations**

***Plotting of Board size vs Percolation rate***

1. **Conclusion**
2. **References:**
3. Dr. Z H Duan class notes on cs.uakron.edu.
4. Algorithms text book
5. https://www.cs.princeton.edu/courses/archive/fall10/cos226/assignments/percolation.html