ACTIVITY 7:

0 1 2 3 4 5 6 7 8 9

[9-0, 3-4, 5-8, 7-2, 2-1, 5-7, 0-3, 4-2]

QUICK-FIND

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 5, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 8, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 8, 6, 2, 8, 0]

[0, 1, 1, 4, 4, 8, 6, 1, 8, 0]

[0, 1, 1, 4, 4, 1, 6, 1, 1, 0]

[4, 1, 1, 4, 4, 1, 6, 1, 1, 4]

[1, 1, 1, 1, 1, 1, 6, 1, 1, 1]

QUICK-UNION

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 5, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 8, 6, 7, 8, 0]

[0, 1, 2, 4, 4, 8, 6, 2, 8, 0]

[0, 1, 1, 4, 4, 8, 6, 2, 8, 0]

[0, 1, 1, 4, 4, 8, 6, 2, 1, 0]

[4, 1, 1, 4, 4, 8, 6, 2, 1, 0]

[4, 1, 1, 4, 1, 8, 6, 2, 1, 0]

WEIGHTED QUICK-UNION

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[9, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[9, 1, 2, 3, 3, 5, 6, 7, 5, 9]

[9, 1, 2, 3, 3, 5, 6, 7, 5, 9]

[9, 1, 7, 3, 3, 5, 6, 7, 5, 9]

[9, 7, 7, 3, 3, 5, 6, 7, 5, 9]

[9, 7, 7, 3, 3, 7, 6, 7, 5, 9]

[9, 7, 7, 9, 3, 7, 6, 7, 5, 9]

[9, 7, 7, 9, 3, 7, 6, 7, 5, 9]

WEIGHTED QUICK UNION WITH PATH COMPRESSION  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[9, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[9, 1, 2, 3, 3, 5, 6, 7, 5, 9]

[9, 1, 2, 3, 3, 5, 6, 7, 5, 9]

[9, 1, 7, 3, 3, 5, 6, 7, 5, 9]

[9, 7, 7, 3, 3, 5, 6, 7, 5, 9]

[9, 7, 7, 3, 3, 7, 6, 7, 5, 9]

[9, 7, 7, 9, 3, 7, 6, 7, 5, 9]

[9, 7, 7, 9, 3, 7, 6, 7, 5, 9]

ACTIVITY - 8

There are four different algorithms : Quick-find, Quick-union, Weighted quick union, Weighted quick union with path compression. An effective technique among these four are Weighted quick union with path compression due to its less time complexity whose formula is M + N log\*N where the time taken to solve the problem is 6 seconds. We can also solve with less iterations and less array access.