**CPSC 6109:** [**Advanced**](https://colstate.view.usg.edu/d2l/lp/ouHome/home.d2l?ou=1218642) **Algorithms**

**Spring 2018**

**Student: Lu Lin**

**Assignment #6**

**Due: 11:59 PM on Sunday, April 8**

Do the following exercises/problems in the textbook. Each problem is worth 50 points with a total of 100 points.

1. Exercise ***21.1-3*** on page 564. During the execution of CONNECTED-COMPONENTS on an undirected graph G = (V, E) with k connected components, how many times is FIND-SET called? How many times is UNION called? Express your answers in terms of |V|, |E|, and k.

Solution:

As per CONNECTED-COMPONENTS line 3 to 5, FIND-SET method is called twice every time the edge in the graph is to be determined; we have |E| edges, thus FIND-SET is called 2\*|E| times.

At the very first we have |V| sets. According to textbook, “Since the sets are disjoint, each UNION operation reduces the number of sets by one.” From given problem we know only k sets remains finally. Thus the UNION operation is |V| - k times, that is, the UNION is called |V| - k times.

1. Exercise ***21.2-1*** on page 567. Write pseudocode for MAKE-SET, FIND-SET, and UNION using the linked-list representation and the weighted-union heuristic. Make sure to specify the attributes that you assume for set objects and list objects.

Solution:

Type definition:

E: an element within a Linked List with three fields:

next: a pointer to next element

value: its own value

set: a pointer to its set

S: set with two fields:

head: a pointer to the first element of the linked list

tail: a pointer to the last element of the linked list

MAKE-SET(x)

element = new E

set = new S

element.next = NIL

element.set = set

element.value = x

return set

FIND-SET(x)

return x.set

UNION(x, y)

s1 = x.set

s2 = y.set

s1.tail.next = s2.head

s1.tail = s2.tail

z = s2.head

while z != NIL

z.set = s1

z = z.next

end while

return s1