

Name: Georges Hatem

CS590 Homework 6: Union-Find
Structures Creativity Exercises

Due Date: March 6, 2022

Problem 7.5.10:

Let's modify the list-based implementation so that this method runs in $O(1)$ time (we want the method `remove(e)` to run in $O(1)$ time and not the whole process, so we can do the following):

Algorithm Remove (S, e)

Input A set, S, that contains the element e

Output return Set, S, with element e removed from it

```
for each x in S do
    if find(x) == e then
        t ← x.element
        (x.prev).next ← x.next
        (x.next).prev ← x.prev
        x.prev ← null
        x.next ← null
    return t
```

This Algorithm above runs in $O(n)$. However, the removal method (which is the method inside of the `(if find(x) == e)` statement) is implemented as union-find structure list-based implementation that runs in $O(1)$ time.

Problem 7.5.12:

In order to make an efficient method for converting A into a set, we better use selection sort. However, in this problem, we are not going to sort the values in a set, but we need to remove any duplicated value. Therefore, selection sort or insertion sort will not be used.

We will do the following:

Array A is the array needed to be converted to a Set. To do this, we will have to use $O(n^3)$ as running time. We need to check if any of the elements in the array equal to another

element in the array and remove the duplicant if any. We will use the Algorithm below:

Algorithm removeDuplicate(A)

Input: remove duplicate elements in Array A

Output: return Array A as a set (without elements duplicated in it)

```
for i  $\leftarrow$  1 to n do
    for j  $\leftarrow$  i+1 to n do
        if A[i] == A[j] then
            for k  $\leftarrow$  j to n-1 do
                A[k+1] = A[k]
```