Homework Assignment #4

Due: Feb 27, 2020 5:30 PM

Written by Jiabao Shen and Jiahong Zhai

Question 1

a. Algorithm

We use forest structure for disjoint set(union/find) with weighted union and path compression.

Step 1: Create two new empty list LS and LD. Loop through L, if encounter S, then put it into LS, if it's D, then put it into LD.

Step 2: Create a new list LF. For 1 to n, make it a disjoint set node and put it into LF.

Step 3: Loop through LS. For each S(i, j) we do the following: a = find(i), b = find(j). If a != b, then union(a, b)

Step 4: Loop through LD. For each D(i, j) we do the following: a = find(i), b = find(j). If a = b, then return "ERROR FOUND" and terminate the program.

Step 5: return k = the size of LF

b. Worst-Case Time Complexity

Step 1: Loop through a list with m elements, operate constant steps for each element. So, it's O(m)

Step 2: From 1 to n, operate constant steps for each number. So, it's O(n)

Step 3 and 4: Since there are n disjoint set nodes at the beginning, so, we have at most n-1 union operations. For each S(i, j) and D(i, j) it takes 2 find operations, so, in total, there are 2m find operations. So it's O(2m log* n)

Step 5: For worst case analyze, we assume all bones are from different species. So, we need to count n elements, it's O(n)

O(m) + O(n) + O(2m log* n) + O(n) = O(m log* n), its better than O(mn)