# Restoring Rainbow Roads

Madhav Tadepalli, EE23B047

September 2023

## Question 1: Using Simple Array

This method uses an array where the index correspond to planet numbers and the values of the index correspond to the set they belong to. This has an average time complexity O$n^2$. This method works well when the number of planets are low (taking ), but is incredibly slow when the number of planets and requests increase (taking 26min for the test case of 1e6 planets and 1e6 requests).

| No of Planets | No of Requests | Runtime(ms) |
|---|---|---|
| 10 | 7 | 1 |
| 100 | 100 | 1 |
| $10^3$ | $10^7$ | 1480 |
| $10^5$ | $10^5$ | 14496 |
| $10^6$ | $10^6$ | 1559742 |

Question 1: Time taken to run the program on average over 2 runs

## Question 2: Worst Case for Question 1

The worst case occurs when a path must be built for every request and the maximum number of array values must be changed in each request. For my program in particular, this would happen if the paths are built starting with the last 2 planets and continuing to build until a path between the first two planets is done.

## Question 3: Quick Weighted Union Find Using Structs

This method creates a struct for each planet, which contain a weight and a pointer to link upward in a tree. This has an average time complexity O(nlog(n)). This method works really well with almost any size of data set, it struggles with a very large number of requests as it must traverse to the head each time even if all the paths are formed.

| No of Planets | No of Requests | Runtime(ms) |
|---|---|---|
| 10 | 7 | 1 |
| 100 | 100 | 1 |
| $10^3$ | $10^7$ | 1512 |
| $10^5$ | $10^5$ | 45 |
| $10^6$ | $10^6$ | 465 |

Question 3: Time taken to run the program on average over 2 runs

## Question 4: Quick Weighted Union Find Using Arrays

This method creates 2 arrays, one being the root array and one being the weight array. This has an average time complexity O(nlog(n)). This method has the same merits and demerits as the method using Structs. The problem can be further optimised using path compression but I have not done that here.

| No of Planets | No of Requests | Runtime(ms) |
|---|---|---|
| 10 | 7 | 1 |
| 100 | 100 | 1 |
| $10^3$ | $10^7$ | 1607 |
| $10^5$ | $10^5$ | 22 |
| $10^6$ | $10^6$ | 487 |

Question 4: Time taken to run the program on average over 2 runs