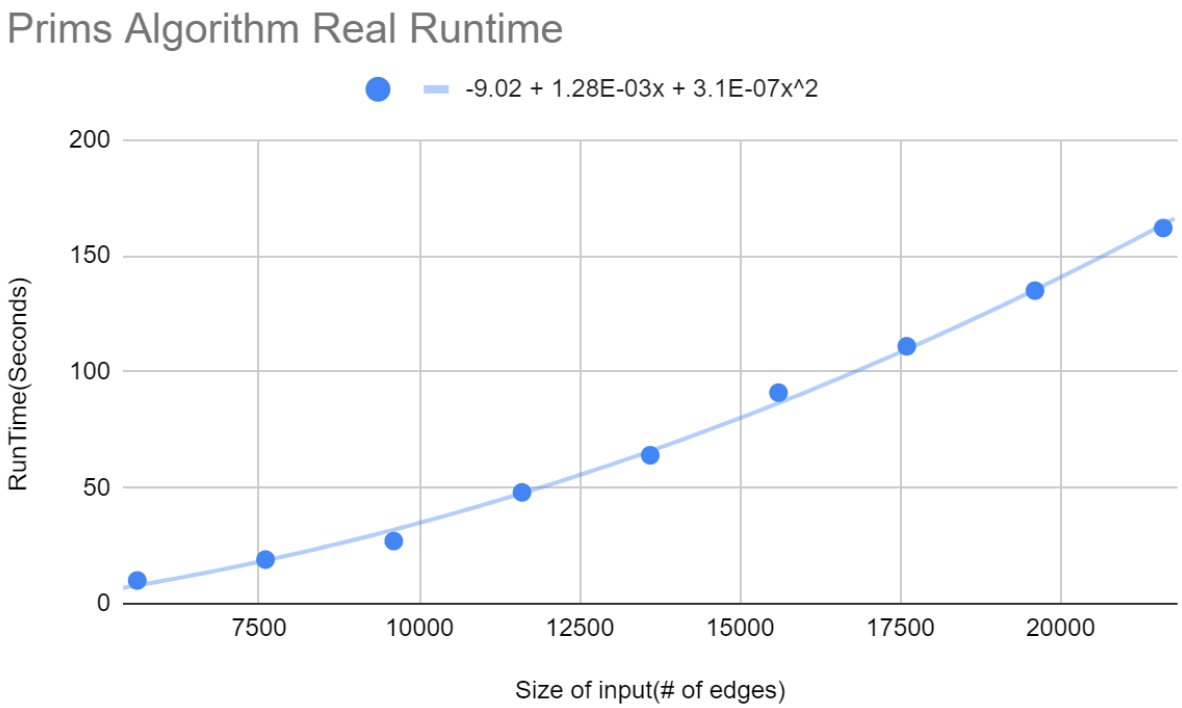
James Howe, Simon Turner, Christina Steinberg, Jackie Quinlivan
12/3/2021
CSCI 311, Talmage

Project 2, city connections

Prim writeup

Time Complexity - 'prim' function: The for loop on line 56 will operate a total of V times for an O(V) runtime, as it checks the set of vertices. The for loop on 64 will also execute V times for an O(V) runtime as it runs through every element in the set of vertices as well. The for loop on line 68, however, will run V^2 times as it is going through all vertices in the set of vertices. With the for loop on line 81, accounting for the set of keys in the parent list, operating a total of V times, the total runtime for this algorithm would be O(V^2). This operates as expected without heap implementation.

Prims runtime chart

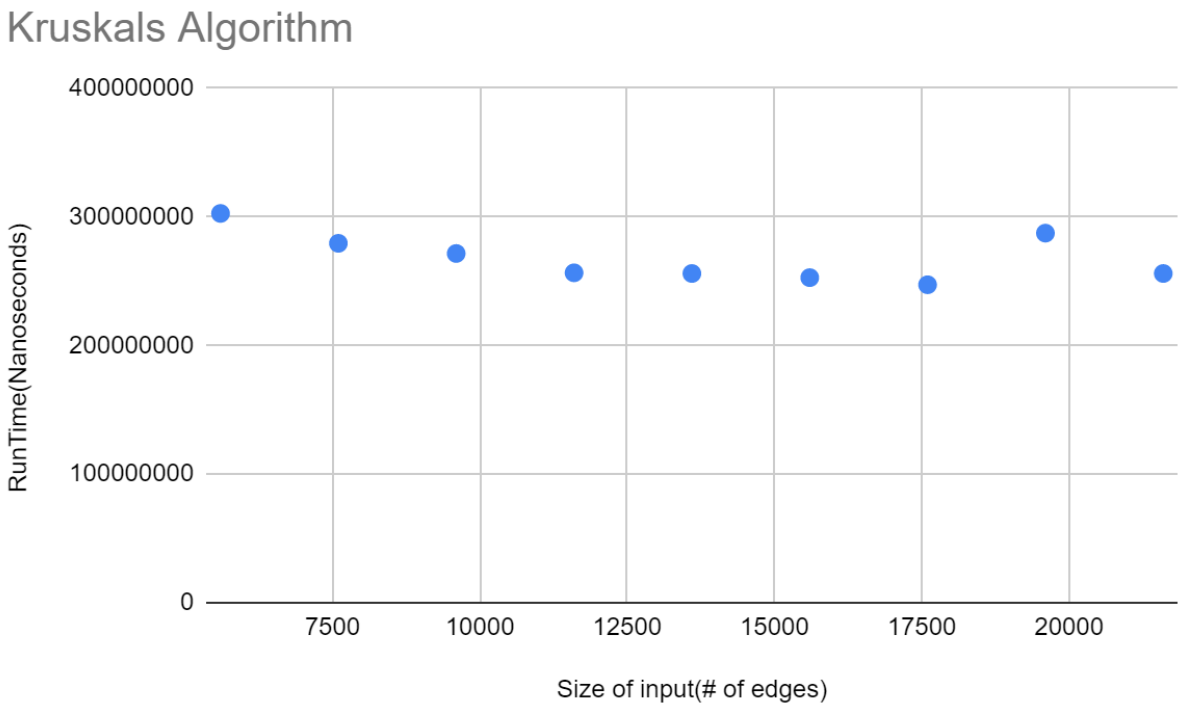


The trendline here is a polynomial, which makes sense for the expected runtime

Kruskal's Writeup

Kruskal's implementation of a minimum spanning tree starts by sorting all the edges in ascending order of their weight. Since Kruskal's is a Greedy algorithm, the greedy choice is to pick the smallest edge. Then, check if it forms a cycle with the current spanning tree, using the union function. If a cycle is not formed, include this edge. Otherwise, release it. Repeat the steps that occur after sorting the edges, until there are (V-1) edges in the spanning tree. The time complexity is O(ElogV). Sorting the edges takes O(ElogE) time. Thenm, iterating through all edges and applying the union

algorithm takes O(logV) in the worst case. So, O(ElogE + ElogV) time. The value of E can be at most $O(V^2)$ and O(logV) and O(logE) are equal. Therefore, the overall time complexity is O(ElogE) or O(ElogV).



No real trend here, it's very fast, it's about the growth you'd expect from O(ElogV) over this range.

Comparison of runtimes

There's absolutely no competition here, the implementation of kruskal's algorithm blows the implementation of prims out of the water, waiting for the data to graph this was a really good way to *feel* how much of a difference these runtimes make and why proper implementation for these algorithms matters a great deal.
Small things can be done to make prims better of course, but the only thing which would make a significant difference(i.e. change the time complexity) would be to make a proper implementation using a priority queue and heap(Should be noted, that this was Attempted, and Failed).