**Análisis Temporal Reto 3**

Requerimiento 1: prueba con ciudad “las vegas”

* Small: 2.06 ms
* 10pct: 3.16 ms
* 50pct: 8.18 ms
* Large: 15.07 ms

Obtener la ciudad tiene orden o(log(n)) donde n es el número de ciudades, en el peor caso también es el número de avistamientos. Ordenar la lista de los n avistamientos en la ciudad tiene orden o(nlog(n)), y en el peor caso todos los avistamientos están en la misma ciudad.

**Conclusión**: el orden del requerimiento es o(nlog(n))

Requerimiento 2: prueba del rango “30.0 a 150.0”

* Small: 2.96 ms
* 10pct: 9.44 ms
* 50pct: 29.69 ms
* Large: 39.45 ms

En el peor caso posible, se deben obtener todas las duraciones del mapa, y al obtener la lista con todos los avistamientos se obtiene un orden lineal, o(n). Lo bueno del algoritmo es que, al hacer la lista, la hace in-orden siguiendo el mapa ordenado (nuevo) de duraciones, por lo que la lista ya está ordenada por duración.

**Conclusión:** el orden del requerimiento es o(n).

Requerimiento 4: prueba del rango “1945-08-06 a 1984-11-15”

* Small: 4.69 ms
* 5pct:
* 10pct: 21.48 ms
* 20pct:
* 30pct:
* 50pct: 58.58 ms
* 80pct:
* Large: 87.68 ms

**Conclusión:** Al igual que en el requerimiento anterior, tenemos un o(n), pues al dar la lista total se agrega uno por uno.

Requerimiento 5: prueba del rango “Latidud: 31.33 a 37.00”, “Longitud: -109.05 a

-103.00”

* Small: 3.02 ms
* 10pct: 9.75 ms
* 50pct: 12.57 ms
* Large: 17.31 ms

En este caso, el algoritmo dabe buscar sobre dos árboles. En el peor caso se requiere buscar entre todas las latitudes y longitudes posibles, pero igualmente lo que toma más tiempo es generar la lista de avistamientos, que nuevamente tiene complejidad o(n).

NOTA: este trabajo fue realizado individualmente.