
Deliverable

By Nussara (Firm) Tieanklin, TCSS

Results

Question 1

The example output from the program.

```
The results coordinates in [latitude, longitude]:
```

```
[47.236220598522706, -122.8645887096292]  
[47.26974981211587, -122.31611909307657]  
[47.27231619593383, -122.28589469943803]  
[47.254490022384076, -122.58358933567983]  
[47.2548835465723, -122.57357166854266]  
[47.25294916808589, -122.60981465067312]  
[47.27742967199156, -122.18820802225255]  
[47.24928293116838, -122.6690233641543]  
[47.24554594904064, -122.72801111040941]  
[47.26386153638679, -122.43027522660769]  
[47.262371494731305, -122.45476091424634]  
[47.29103867738846, -121.91932500830598]  
[47.24545016139363, -122.72614127759584]  
[47.23229519667999, -122.91659100508288]  
[47.23827819312994, -122.82412706321396]  
[47.24301619921194, -122.77046270676172]  
[47.26603327082752, -122.38835989434664]  
[47.292797361501904, -121.87792656518067]  
[47.25668947789633, -122.54089234645532]  
[47.24637149162688, -122.7107292870431]
```

c) Determine the time complexity of the above (part b) algorithm.

The runtime in the current solution is $O(n^2)$. The runtime will increase as the given n number of random coordinates increases. Moreover, we may encounter the scenario that the program will go into the infinite loop in case that we have an incredibly large n while we are only given the *radius* smaller than 400 meters or the minimum distance between any two coordinates because the program will keep random the coordinates endlessly.

d) Is it possible to generate a similar algorithm having time complexity of $O(n)$?

I however think that this program will not be able to complete with the time complexity $O(n)$ because the runtime is completely dependent on the user input n of the numbers of outputs where we could not expect that the given input could be. Since one random coordinate covers $\pi(0.4)^2 \text{ km}^2$ within the certain restricted area by the given *radius* of

$\pi(radius)^2 \text{ km}^2$. If the n is incredibly large with the *radius* smaller than 400 meters as was stated in part b), we will have larger and larger runtime and finally may enter into the infinite loop.

