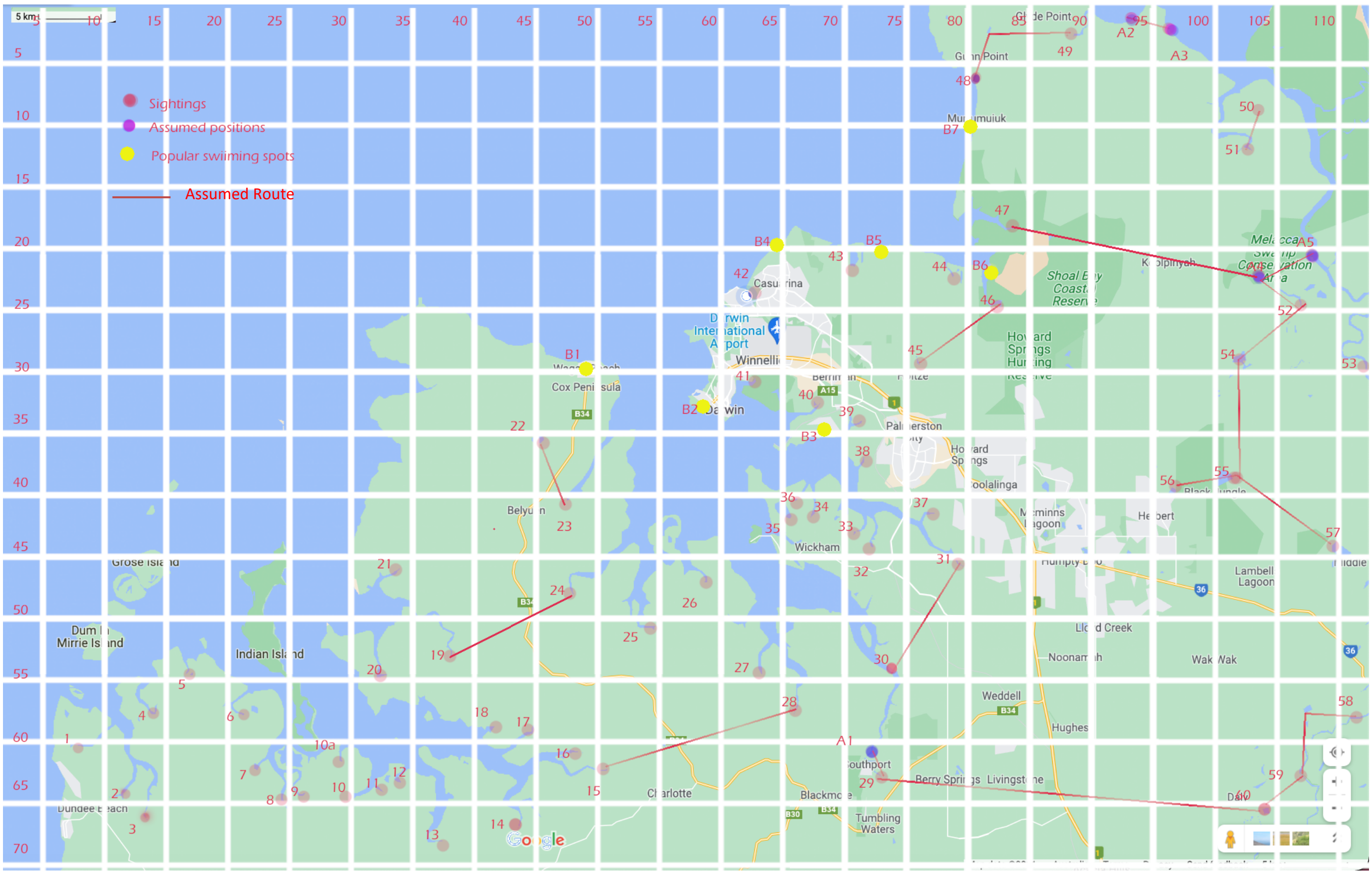
Crocs: Optimizing Croc culling and removal



For this assignment you will be given a csv file with:

1. Location of Croc sightings over the years
2. Paths of croc travel between sightings
3. A draft python file to fill in the code, or replace with a related Java/C file (or enter pseudo code)

1 Preparation

This exercise is based on created data, but the context is relevant. There are much less sighting than used in this exercise

Information: Crocodiles are often sighted around the Darwin area. Some of these are quite historic but they show where crocodiles have been breeding and feeding. They can travel down rivers, around bays and we have reason to believe they travel across areas of land marked with red trails. The exact route of these trails is not known, we are providing the best estimate given sightings of croc at different times in this area.

The red dots are sightings, usually associated with breeding locations, as the croc is less mobile at this time so more likely to be tracked. Sighting can be through finding fresh tracks or sightings of the animal in the location, which will include the number of sightings over a season.

Your first task is to process the data files and populate the variable *locationList*, with the data of the graph within the python file. You can calculate the travelling distance between two locations using the most direct path between two geo-locations, hence it is only an approximation of the actual distance between two locations.

Some extra points are provide as assumed stops on the transit routes of the croc when travelling between water courses. Otherwise points are joined along rivers and around the coast. Neighbours are only given in one direction the opposite direction is assumed to exist also.

Requirements

Make sure your python/pseudo code file has the assumptions you have made in interpreting the clients' requirements. This is so that if you code does not work as I expect, I can check if you code works according to your requirements you have assumed, then you will receive more marks than if your code works according to requirements that are not reasonable.

Note: Distance ***along a single path between adjacent points*** *can be estimated as* ***distance between points,*** *the complete track on the map around shoreline etc is not needed*

Question 1

To test the possibility of blocking route between two locations, the rangers have to perform an exhaustive search on the path options between the two locations*, along all direct paths in the same region of map.* This will involve them moving between known sightings and along estuaries or shoreline to assess the ability for crocs to pass, or possible ways to block their passage.

Your task is to devise an algorithm to determine the minimum cost of performing an exhaustive search along all possible paths between two points where this cost is proportional to ground covered. Also give the single list of sequential point locations that were chosen to *segment* the route in order to obtain this minimum cost. If you find two routes from start point to the end point you need to include both in the optimum path. The cost is estimated just in units (how many units of work time required). Routine is *computeCosting (location1, location 2).*

Question 2

The engineers are now conducting a study on the benefit of constructing a croc barrier **at an existing location**. One important data that they need is a comparison between the *current distance* between two locations using existing trails, and the *new alternative distance* between two locations if there is a blockage in the present shortest route between them. For instance, a monitoring device could be set up to alert if a croc passes the *blocked point* and have a crew sent in on sightings. These devices are expensive, and risk being stolen, so we want the optimum place on the path to locate this.

It is assumed it is not worth blocking overland points marked as **assumed** as these can be wide ranging in their actual location

For any two locations or group of locations, the higher the ratio between current distance and *new alternative distance* is, the more benefit can be obtained by building a blockage **at that point**. You should return the ***ratio of this improvement*** and the ***point that will be blocked*** to achieve this using method improveDistance(a,b)

Do not block the starting or end point of the path, that is too easy

Question 3

You are again asked to find the minimum distance between two locations in terms of number of metres and hence time for croc to travel. Croc speed is about 16 km/hr in water and 6km/hr on land. You are required to specify the route in terms of the points travelled through on the path. Method minTime() returns an array and a time value

As an extension, provide the number of crocs in a certain radius x miles of a beach. Using this array of locations, decide which is the **optimum path point** to insert a blockage that would make the beach safer, by increasing the time the maximum number of crocs would have to travel. **Route to beach is through any point on coast nearest to beach**

ChangeLog 14/10/21. Clarified Question 2 to not use assumed routes and provided key on map

Fixed data table to put x,y in correct columns

Include assumptions in your code file

16/10/21. Expanded CrocMonitor class to upload data

Example outputs provided

Change requirement for Qu1 and 2 in bold

18/10/21. Corrected example in Python file and method computePathDistance now acts on path not end points (a,b)

Added more examples and description on methods

Updated data files

Explained path for extension to safety at beaches

Note: Do not remove description from python file, they give the requirements. Just expand on that if you need

21/10/21 Added more examples in code

Added description of how to consider scope of Qu1 path, in method findScope

Added description of this in Qu1

Put restriction on Qu2 to not block at start or end point (trivial)