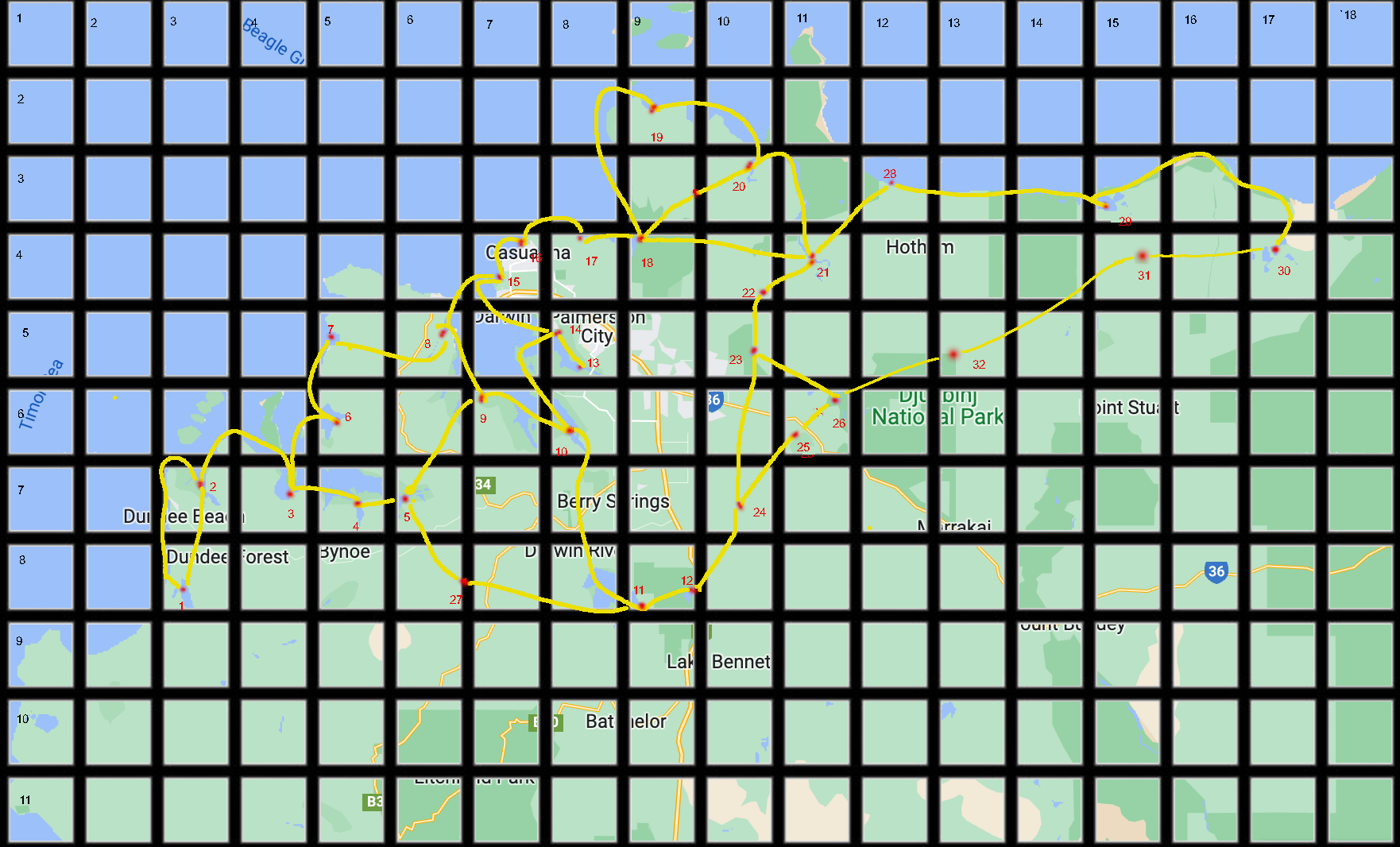
**Crocs: Optimizing Croc culling and removal**



For this assignment you will be given an expanded csv file with:

1. Location of Croc sightings over the years with numbers
2. Paths of croc travel between sightings

You will upload to the submission link a python file or a related Java/C file or a word document with 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 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 in Assignment 2 was to process the data files and populate the *locationList*, with the data of the graph within the python file.

You calculated 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.

Points are joined along rivers and around the coast or across land. The terrain is data you should store. Neighbouring points are only given in one direction in the xls the opposite direction is assumed to exist also.

***We now give you data on number of croc sightings per location to add***

This may need to redesign the type of data structure you chose

***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 with no explanation*.*

*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***

We are sending out teams to cull crocodiles. We will send them out to the sites we know mostly commonly have crocodiles. As we are sending out limited team, we want to go to sites near each other with the same likelihood.

Also you should realise that we will be updating the sighting data at times as new ones found, both sites and sightings.

1. Using your array of data points and their weighting now provided for number of sightings, provide code or pseudo code for below methods:

Note: Marks awarded for correct working or logical answer, plus for most efficient method

Note: The method is called first with previous\_site=null.

Provide

Next\_site(previous\_site, previous\_number):

Find next site to check for crocodiles

This will output sites in terms of max previous number of sightings and then same numbered sites at locations nearest that site

Return location on map and site number

1. If a croc is found we need to update the data

Provide

Add\_sighting(site: location, water[Boolean], update[boolean]):

Add sighting to graph

Include location, and if\_water

Update used if previously in database, so just update number of sightings

1. You will need a sort action that can be called by Next\_Site or Add\_sighting. Give reason for choice of which one calls this function. Give reason for sort chosen

Provide

Sort\_data():

Sorts data for finding next site to check