**Flock of Sheep (TTC 2hrs):**

We have the coordinates of a number of sheep in a large field. Propose an algorithm that can determine how many groups they form. What are the advantages and disadvantages of the proposed algorithm? Can you describe any circumstances in which it can go wrong, and if so, how?

A blue and purple circles and crosses

Description automatically generated

I started by visualising the field, the sheep and defining a group. When I thought about a group, I considered a group of people, and you can normally tell a group of people by their physical proximity to each other. As such, there is some arbitrary distance, **D**, which we use to determine if someone is a member of a group.

As we know the coordinates of the sheep, we can calculate the Euclidean distance, **E**, between sheep which we can then compare against **D**. If **E** is less than or equal to **D**, then we can consider them a group.

Therefore, we can start with a particular node and check the Euclidean distance to the other nodes. If the distance to the other node is <= **D** then we group them, and tick the node as being visited. As a preliminary thought, I would then store them in a Dictionary type structure, where the first visited node is the key, then adds all nodes visited within **D** to a list as the value.

Once no nodes remain that can be visited in a distance <= D, we start searching again from a new unvisited node. Once no unvisited nodes remain, we can get the number of keys that hold a value in the dictionary, as that will show the number of groups.

Disadvantages:

* As this is a brute force method, the time-complexity is at worst case scenario O(n^2). This is highly inefficient and will scale badly into larger searches.
* If an unvisited node is close to another node that is a member of a group, but not close to the initial node that formed the group then it won’t be included in the group. It won’t expand the logical group sizing.

Advantages:

* Simple and intuitive algorithm.
* It also makes no assumptions about the data, so can be easily edited/evolved to become more efficient. (E.g. a bit of research reveals that you can partition the space in the grid into cells, and associate points to the cell or neighbouring sells within a fixed radius)

I couldn’t think of any circumstances where it could go wrong within the 2-hour timeframe.