# Project 5: Wolf Tracking

## Background:

Lydia graduated from Denison in 2020 and now works for the National Park Service at Yellowstone National Park. She is part of the wolf rehabilitation project that is reintroducing wolves to the park. Her job is to track various wolf packs in the park via radio collar devices which have been attached to several of the pack members.

The radio collars communicate with GPS satellite devices and send data signals to various radio antennae installed around the park. Lydia can get real-time location information for all six of the known pack families that inhabit the park. For every 24-hour period, Lydia dumps all the signal information into a separate text file for each wolf and then processes these data files the next day. Each line of the file has three datapoints:

            hh:mm            x          y

where the timestamp indicates the time of each signal (24-hour format with minutes), and the latitude and longitude (x and y) indicate the approximate geographic location of the wolf. The time stamps are taken every 15 minutes and thus there are (24 \* 4) 96 data points in each file. The first timestamp is 00:00, then 00:15, 00:30, ... and the last timestamp is 23:45.

The GPS locations are translated into a grid that is overlaid onto the park. Each x and y coordinate is an integer in the range of 0 to 499 inclusive.

Lydia’s current research project is to track unusual exploration behavior by the collared wolves. In some circumstances, the wolves seem to complete an out-and-back cycle where they will follow a route and then backtrack on that exact same route in reverse. In the data files, this will appear as a series of GPS datapoints, a turn around point, then the exact same sequence of GPS points in reverse order. This sequence may have an even number of datapoints (with a repeated pair in the middle) or an odd number of datapoints (with a single datapoint in the middle). Lydia is looking for these types of sequences in the dataset.

## Program Logistics:

You are to process data files of this format, looking for sequences of out-and-back tracks. Specifically, you want to find the longest out-and-back sequence as measured by elapsed time. You are to report the total elapsed time in any 24-hour period for the longest out-and-back sequence recorded during that day.

You are given two sample data files. The correct responses from each of those files is shown below:

File:                             wolf1.txt         wolf2.txt

Expected output:        00:45               01:00

where these two particular wolves had a longest out-and-back track that lasted 45 minutes and 1 hour respectively. You can find these longest sequences in the files by examining timestamps 15:30-16:15 or 16:15-17:00 in wolf1.txt and in 01:30 - 2:30 wolf2.txt.

Your program is to read from standard input (using cin). Use a loop to read 96 points. I recommend reading a string, an int and an int for each line of input.

**string    time;**

**int     x, y;**

**cin >> time >> x >> y;**

As in project 4, we will use I/O redirection to send the datafiles to the programs as input:

**cat wolf1.txt | wolf.exe**

**00:45**

You can assume there will always be 96 data points, and they will always be formatted as described above and seen in the two example files. **You do not need to check for erroneous input.**

Store the values into three arrays – an array of strings and two arrays of ints. You will then parse the two arrays of ints looking for these out-and-back sequences. Identify each such sequence, measure the elapsed time (ending timestamp – starting timestamp). Keep track of the longest such out-and-back sequence and print the longest elapsed time.

## Levels of Achievement:

* D range: A non-working (but still well-documented) program. Points are awarded based on progress made.
* C range: A working, well-documented program that mostly correctly reads in the data and finds spans of out-and-back sequences but may fail on some test cases.
* B range: A working, well-documented program that correctly finds all spans of out-and-back sequences.
* A range: A program that achieves the B level of correctness but does so in a more efficient way. Specifically, find a solution that is more efficient than O(n3). This means finding a solution other than one that checks whether each possible combination of start and end points is an out-and-back sequence.

## Learning Goals:

• Practice with C++ data and arrays  
• Complex algorithm development  
• Practice with both iterative improvement and functional decomposition

• Practice creating one’s own test suite