Combinatorics Practical 2

Implementing A* search

In this practical you will implement an A* search to find routes through a grid.

The grid will be provided as a 2D matrix of numbers, where each number is the height at that square, like this:

```
[
    [1,1,1],
    [1,2,3],
    [2,1,1]
]
```

You should always find a route from the top-left square to the bottom-right square.

At each square you can move up, down, left or right, as long as you do not leave the grid.

The time taken to walk to from a square of height a to a square of height b is: $1 + (a - b)^2$.

You will be given some basic framework code. You have to fill in `Graph` and `find_shortest_path`. You should implement the `Graph` class first, and then use this to implement `find_shortest_path` using the A* algorithm.

The code includes a `print_path` function, which provides a way of visualising your path. It uses letters to show height (0 is a, 1 is b, etc). Positions in the path are marked in a capital letter.

The code currently implements vertices as a pair, `(x,y)`. You are welcome to change this, or any other code, if you wish.

Your code when run should perform the following steps:

- Ask for a filename of a height grid
- Let the user pick which heuristic to use. Option "1" should choose your best heuristic.
- Perform an A* search, printing out:
  - The path length
  - The number of nodes explored during search
  - The path itself

You must implement at least two heuristics.

One heuristic must be the "as the crow flies" "straight-line" heuristic, ignoring the heights.

You should also design and implement at least one other heuristic. You will be graded on the quality of your heuristic (where quality is measured by the smallest size of search). Submit a brief report, proving your heuristic (or heuristics) are admissible, and discuss how your heuristic (or heuristics) compare with the “straight-line” heuristic.

You may use or expand existing published heuristics, however you must reference the other heuristics if you do this.

Your code should be clear, and well documented in English.

Some height maps are provided. You may create other height maps, and you are encouraged to share height maps with other students.

Your grade will be calculated as:

- 50% - Correctness of code.
- 20% - Clarity of code.
- 30% - Quality of report.