

COMP1730/6730 Assignment Description

Matthew Alger and Jeffrey Fisher

16-05-2019

Important

- The assignment is due early in week 12 - Tuesday, May 28th, 11:55 PM.
- The code section of the assignment can be completed in groups of 3. These groups must be registered on WATTLE by Wednesday of Week 10 (May 15th).
- The report section of the assignment **must** be completed individually. You can share ideas (with proper attribution) but your write-ups must be your own.
- Include your university IDs in every file you submit.

Introduction

Geographic data is useful for a wide variety of purposes, from urban planning to environmental management. In this assignment you will work with a real elevation dataset of the Australian National University. These data were taken in 2015 by the ACT as part of a [LiDAR scan of the entire territory](#). The University looks a little different now, but hopefully you can still recognise some places!

You will have to complete a number of programming questions, and write tests for your solutions. You will also have to think about what you've written, and write a report about your work.

Requirements, Expectations, and Marking Criteria

You need to submit three files: `assignment.py`, the Python script containing your assignment submission; `test_assignment.py`, the Python script containing your tests; and `report.pdf`, a PDF version of your report.

For your code, we have the following requirements and expectations. **If you don't follow these requirements, you may receive 0 marks.**

- Your code must be syntactically correct.
- You cannot import any modules except `math`, `numpy` and `matplotlib`.
- You shouldn't change the arguments or names of functions in the template (though you can add new functions if you wish).
- You should not use any global variables or have any code outside of your functions unless it is in the `if __name__ == '__main__':` block.
- Your code shouldn't raise any unintentional exceptions or warnings.

We will mark your code based on correctness and quality. "Correctness" means that your code has to answer the questions in this assignment. "Quality" means your code is readable, good, and efficient:

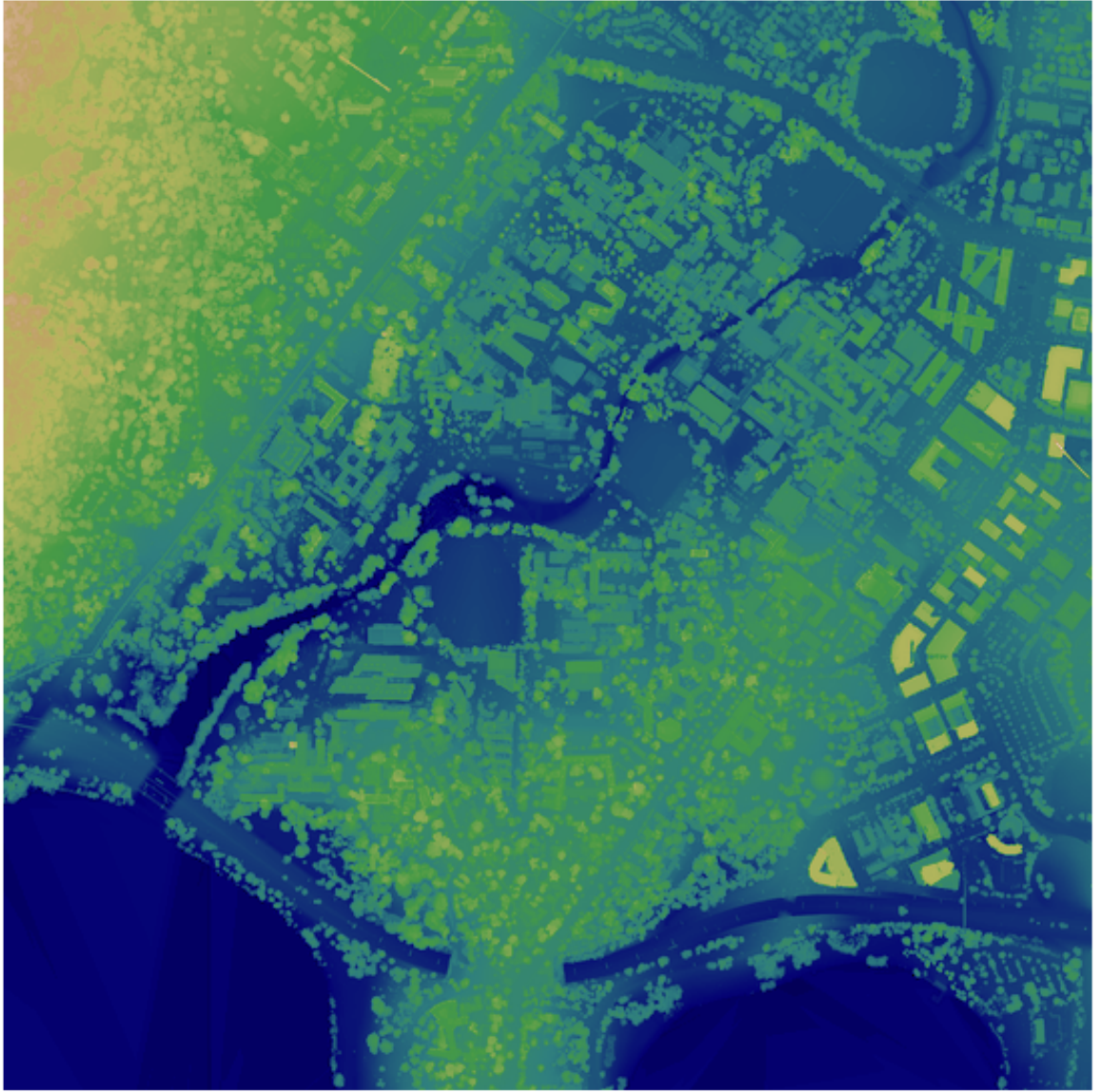


Figure 1: ANU heightmap.

- You should use docstrings and comments where appropriate.
- Your function and variable names should make sense and be descriptive.
- You should use suitable data types to solve problems.
- You should organise your code appropriately, using functions where necessary.
- Your code should be reasonably efficient: Don't make the computer do too much unnecessary work.

We will also mark your report based on correctness and quality. **If we cannot understand or find your answers in your report, you may receive 0 marks.** The report should be:

- clear (it should be easy to find and understand your answers);
- concise (use simple words and sentences);
- well-organised (use headings and subheadings where appropriate);
- relevant to your code; and
- 1–3 pages. **We will stop reading after 3 pages!**

In this assignment, you will have to make some choices on how to design your solution to problems, and you will be asked to justify these choices in your report. You should show understanding of the problem and your solution, and convince your marker that your solution solves the problem in an appropriate way. Much like real life, many questions in this assignment do not have just one correct answer, so it is especially important to justify the decisions, assumptions, and solutions you've made.

The assignment is divided into roughly 50% code functionality, 25% code quality and 25% report.

There are 6 questions with uneven weightings:

- Question 1 - 10 marks
- Question 2 - 10 marks
- Question 3 - 20 marks
- Question 4 - 20 marks
- Question 5 - 20 marks
- Question 6 - 20 marks

Please note that these percentages are indicative only.

The Data

We have given you two datasets. The first dataset is an elevation map of the whole ANU. The second dataset is a much smaller elevation map that covers Burgmann College and the Research School of Physics and Engineering. The small dataset will let you quickly test your code without taking too long.

We've split the elevation data for each dataset into two files: one contains the ground-level elevation, and the other contains the heights of each building in the area.

In total there are four files:

- `height_anu.txt`,
- `buildings_anu.txt`,
- `height_rspe.txt`, and
- `buildings_rspe.txt`.

`anu` is the whole-ANU (large) dataset. `rspe` is the Burgmann/RSPE (small) dataset.

We've given you code in the assignment template that will load these files.

Your code should work quickly on at least the `rspe` dataset. For a good efficiency mark it should also work quickly on the `anu` dataset. We will test your code on some additional data in the same format as the given datasets, but in different places in Canberra.

The Task

You are provided with `assignment_template.py`, which contains the basic functions of the assignment. The functions are incomplete. In this assignment, you will fill in the blanks and complete the missing functions, as well as add your own testing functions. You will also write a short report about your functions and decisions.

Question 1: Area Above Water

We've written a simple function for finding how much area is above water in a map in *square kilometres*. However, it doesn't work! Fortunately we've written some tests, so we know that it doesn't work. Without changing the existing tests, fix `cells_above_height` and `area_above_water` to work correctly. Also make sure to add docstrings and comments to our code where appropriate.

Lake Burley Griffin has an elevation of 557 metres and each cell in a map is 1 metre by 1 metre.

The assignment template contains the functions for you to fix:

```
def cells_above_height(heightmap, height):
    n_cells = 0
    for row in heightmap:
        for cell in row:
            n_cells += 1
    return n_cells

def area_above_water(heightmap, water_level):
    return cells_above_height(heightmap, water_level)
```

In your report you must:

- Explain what you changed to make the functions work correctly.
- State how much of each map is above water in *hectares*.

Question 2: Highest Point

Write a function `highest_point` which takes a heightmap and returns the highest point as a tuple of (x, y) coordinates. We have included tests for this function.

The assignment template contains a placeholder function for you to fill in:

```
def highest_point(heightmap):
    pass
```

Using your function, answer the following questions in your report:

- What is the highest elevation on each ground map?
- What is the highest elevation on each map if you include buildings?
- How tall is the tallest building on each map?

Question 3: Image Gradients - Slope and Aspect

As you may know from walking around the ANU, there are a number of hills. In this question you will write a function that calculates the steepness, or *slope*, of each cell in the heightmap. We can use this to create a so-called *image gradient*, where each cell on the heightmap becomes a pixel on an image, with its colour value determined by the slope. You will also be calculating the *aspect* and creating an image gradient for this as well.

We calculate the slope as follows: for each cell, subtract the height of the cell on its left from the cell on its right. This is the *horizontal gradient*. Then subtract the height of the cell above it from the cell below it. This is the *vertical gradient*. Square both gradients, then add them together and take the square root. This is the slope, or *total gradient*.

The aspect of a cell is also based on the values of its neighbours. Letting Δx be the horizontal gradient and Δy be the vertical gradient, the aspect is

$$\alpha = \arctan(\Delta y, \Delta x).$$

You can find the arctan function in the `math` module called `atan2`.

Write two functions `slope(heightmap, x, y)` that takes a heightmap and a position and returns the slope at that point and `aspect(heightmap, x, y)` which takes a heightmap and position and returns the aspect at that point.

We have provided a function `visualise_slope(slope, heightmap)` that will show the slope computed across the whole heightmap. We have also provided `visualise_aspect(aspect, heightmap)` that will show the aspect function computed across the whole heightmap.

The assignment template contains two placeholder functions for you to fill in:

```
def slope(heightmap, x, y):  
    pass  
  
def aspect(heightmap, x, y):  
    pass
```

We have provided some tests for the functions on one of the data sets. However, you should write some additional tests, paying particular attention to *edge-cases*.

In your report, answer the following questions:

- How did you handle the boundaries of the map, and why did you handle them this way?
- Why are your tests good additional tests for these functions?
- You should also include the images produced by running the corresponding visualisation functions.

Question 4: Roller-Blading Championship

Each year, to determine who should be their next president, the ANU roller-blading society holds a championship event. You are considering entering the race, and you want to know what route you should take to maximise your chances of winning.

Your task is to write a function `find_path(heightmap, x, y, water_level)` which takes a heightmap and a starting position, and calculates the path you should follow. Your function should return a list of tuples of the (x, y) coordinates of the cells along the route.

You can assume that if you are in a particular cell, you should move to the adjacent cell with the lowest elevation. You may also assume that you will stop when all adjacent cells have an elevation that is equal to or higher than the current cell, or when you reach the water level.

The assignment template contains a placeholder function for you to fill in:

```
def find_path(heightmap, x, y, water_level=557):  
    pass
```

We have provided a function `visualise_path(heightmap, find_path, x, y, full_map=False)` which will plot a path on the heightmap for you. By default, this function only shows the section of the map where the path is, but you can force it to display the whole map by setting `full_map=True`.

We have also provided some tests for this question, however they are intentionally not exhaustive. You should add some additional tests of your own, paying particular attention to *edge cases*.

In your report you must:

- Describe any additional assumptions you made when solving this problem.
- Explain why your additional tests are appropriate and what *edge cases* you considered.
- If you experiment with some different starting positions, you will likely find that most of the paths you calculate are relatively short. Explain why we might need a more sophisticated approach to solve this problem in a realistic fashion (note you don't actually have to implement a more sophisticated approach, just explain what we should incorporate into the solution).

Question 5: Campus Buildings

`buildings_rspe.txt` and `buildings_anu.txt` contain heightmaps with just buildings in them — that is, no ground. The ground is marked by zeroes on the map. Your task, is to find all the buildings.

Write a function `find_buildings(building_heightmap)` which takes a buildings heightmap and returns a list of sets. Each set should contain the coordinates of a single building (stored as tuples of the (x, y) coordinates of all the cells of the building). Assume that if two non-zero cells are next to each other, then they are part of the same building.

The assignment template contains a placeholder function for you to fill in:

```
def find_buildings(buildings_heightmap):  
    pass
```

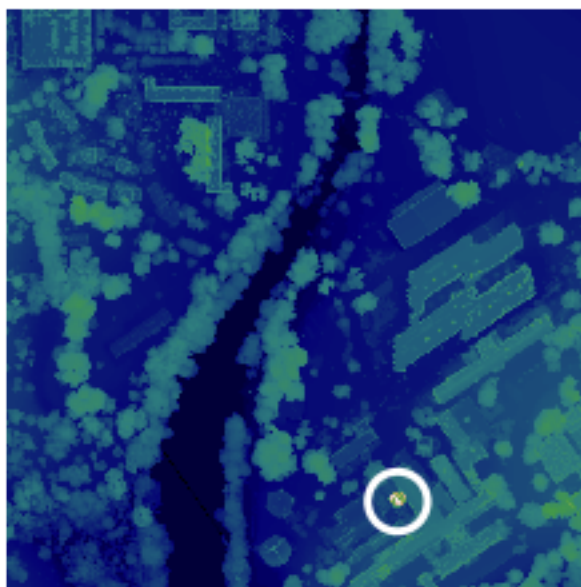
You are also required to implement some tests for this function.

It is up to you how to implement this function. Your code should be reasonably efficient. In your report:

- Describe (and justify) any assumptions that you made in solving this problem.
- Describe your algorithm.
- Write down the time complexity of your algorithm, and explain why this is the time complexity.
- Explain why your chosen tests are appropriate.
- How large is the largest building at the ANU (in square metres), and what is the building.



Figure 2: Image: The Heavy Ion Accelerator Facility. Credit: ANU JCOS



Question 6: The Heavy Ion Accelerator

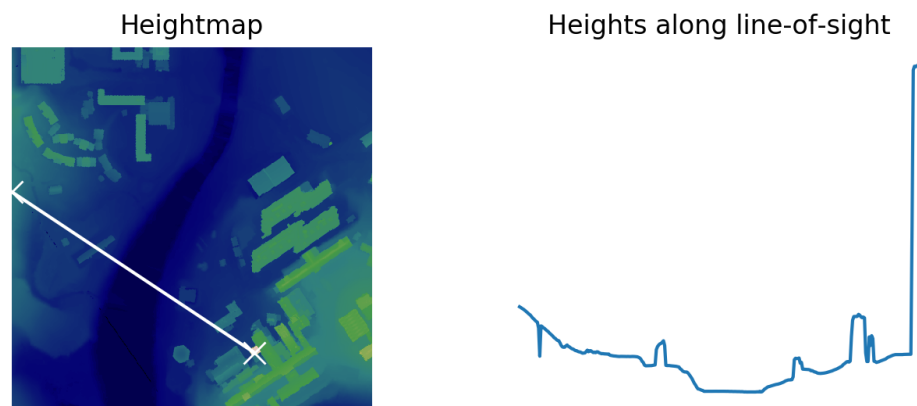
`buildings_rspe.txt` and `buildings_anu.txt` contain heightmaps with just buildings in them — that is, no ground. The ground is marked by zeroes on the map.

The Heavy Ion Accelerator Facility (HIAF) is a particle accelerator facility at the Research School of Physics and Engineering. It's readily visible in the rspe heightmap at (337, 423), circled above:

The HIAF is 12 stories high, making it the tallest building on campus (as you should have discovered in Question 2!). It should therefore be pretty easily visible. In this question you will find all the locations where you can see the HIAF on campus.

Question 6a: Heights along the line-of-sight

Write a function `line_of_sight(ground_heightmap, building_heightmap, x1, y1, x2, y2)` which takes a ground heightmap, a buildings heightmap, and two coordinates (x_1, y_1) and (x_2, y_2) , and returns a list of heights between each position. We have provided a function `visualise_line_of_sight(ground_heightmap, building_heightmap, x1, y1, x2, y2, line_of_sight)` that will let you see if your function works. For example, on the RSPE map, `visualise_line_of_sight(ground_heightmap, building_heightmap, 0, 200, 337, 423, line_of_sight)` should show you a plot like this:



The assignment template contains a placeholder function for you to fill in:

```
def line_of_sight(ground_heightmap, building_heightmap, x1, y1, x2, y2):  
    pass
```

Question 6b: Is the HIAF visible?

Write a function `is_hiaf_visible(ground_heightmap, building_heightmap, x, y, hiaf_x, hiaf_y)` which takes a ground heightmap, a buildings heightmap, an x and y position, and the x and y position of the HIAF, and returns a boolean: True if the HIAF is visible, and False if not.

The assignment template contains a placeholder function for you to fill in:

```
def is_hiaf_visible(ground_heightmap, building_heightmap, x, y, hiaf_x=337, hiaf_y=423):  
    pass
```


You are also required to implement tests for your functions.

It is up to you how to implement this function. Your code should be reasonably efficient. In your report:

- Describe (and justify) any assumptions you made when solving this problem.
- Explain your algorithm.
- Write down the time complexity of your algorithm, and explain why this is the time complexity.
- Explain why your chosen tests are appropriate.
- Does your approach correspond accurately with reality? If not, why is it different?

Please note that this is intended to be a harder question. Make sure you have the other questions working before spending too much time on this question.

The Report

Alongside your code, you also have to submit a written report that answers the questions described above. Answers to the questions should be clearly indicated as such.

Submission

- Submit the assignment through the submission link on WATTLE.
- We will mark the most recent submission prior to the deadline.
- Every group member must submit a zip folder containing `assignment.py` (the code), `test_assignment.py` (the tests) and `report.pdf` (the report).
- Your student id and those of your group members should be included in every file you submit.
- The code should be identical for every member of the group.
- The report must be entirely your own work.
- No late submissions will be accepted without prior arrangement with the convenor.
- Students will only be granted an extension on the submission deadline in extenuating circumstances as defined by [ANU policy](#). If you think you have grounds for an extension, you should notify the course convener as soon as possible and provide written evidence in support of your case (such as a medical certificate). The course convener will then decide whether to grant an extension and inform you as soon as practical. Extensions will only apply to the individual reports.

Plagiarism

The only groupwork in this assignment should be on the code. All reports must be individual work. We do encourage you to discuss your reports, but we expect you to do the write-up by yourself. Reports will be considered under the usual individual plagiarism rules. If you are unsure about what constitutes plagiarism, please read through the [ANU Academic Honesty Policy](#).

If you do include ideas or material from other sources (in your code or your report), then you clearly have to make attribution by providing a reference to the material or source in your report. We do not require a specific referencing format, as long as you are consistent and your references allow us to find the source, should we need to while we are marking your assignment.