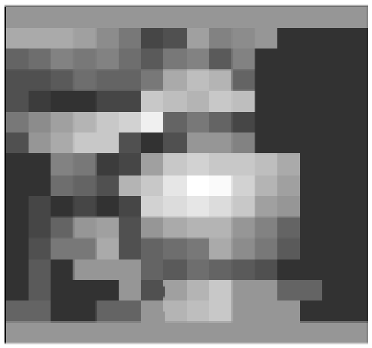
COM1005: Machines and Intelligence
Semester 2: AI Techniques
Assignment 1 2018
The Rambler's Problem

This assignment carries 12.5% of the assessment for COM1005

1. The problem

The problem is to work out the best walking route from a start point to a goal point, given a terrain map for the walking area. A terrain map specifies the height of each point in the area:

A realistic Terrain Map



A simple Terrain Map.
White is highest,
black is lowest.
In **tmc.pgm**

*For a rambler, the best route is the one which involves **the least effort**, which is a combination of how far you walk and how much climbing you do, as defined below.*

In this assignment you'll experiment with search strategies for solving the Rambler's problem.

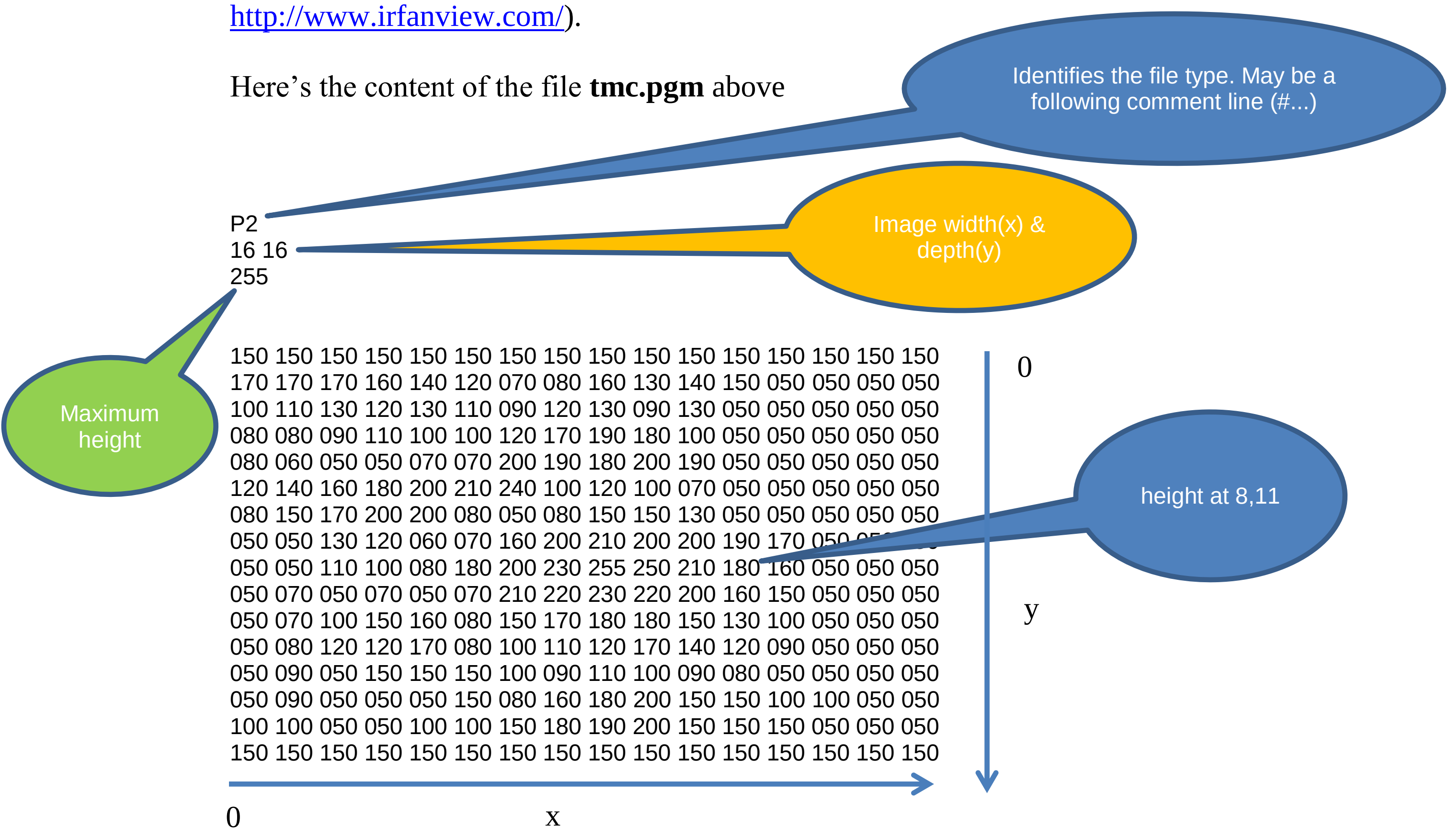
2. Representing Terrain Maps

We'll store our terrain maps in Portable Grey Map (pgm) format (http://netpbm.sourceforge.net/doc/pgm.html).

In this format each cell is represented by an int from 0 to 255.

You can view and edit pgm files in irfanviewer (free download: http://www.irfanview.com/).

Here's the content of the file **tmc.pgm** above


Identifies the file type. May be a following comment line (#...)
Image width(x) & depth(y)
Maximum height
height at 8,11
0
y
0
x

```
P2
16 16
255

150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150
170 170 170 160 140 120 070 080 160 130 140 150 050 050 050 050
100 110 130 120 130 110 090 120 130 090 130 050 050 050 050 050
080 080 090 110 100 100 120 170 190 180 100 050 050 050 050 050
080 060 050 050 070 070 200 190 180 200 190 050 050 050 050 050
120 140 160 180 200 210 240 100 120 100 070 050 050 050 050 050
080 150 170 200 200 080 050 080 150 150 130 050 050 050 050 050
050 050 130 120 060 070 160 200 210 200 200 190 170 050 050 050
050 050 110 100 080 180 200 230 255 250 210 180 160 050 050 050
050 070 050 070 050 070 210 220 230 220 200 160 150 050 050 050
050 070 100 150 160 080 150 170 180 180 150 130 100 050 050 050
050 080 120 120 170 080 100 110 120 170 140 120 090 050 050 050
050 090 050 150 150 150 100 090 110 100 090 080 050 050 050 050
050 090 050 050 050 150 080 160 180 200 150 150 100 100 050 050
100 100 050 050 100 100 150 180 190 200 150 150 150 050 050 050
150 150 150 150 150 150 150 150 150 150 150 150 150 150 150 150
```

Code for handling terrain maps is in the Ramblers folder in the java2018 archive you can download from MOLE
You're provided with the file **tmc.pgm** and a java class **TerrainMap** whose constructor is given the filename of a pgm image and reads its contents, i.e.

```
TerrainMap tm = new TerrainMap("tmc.pgm");
```

TerrainMap has the following accessors:

```
public int[][] getTmap() {return tmap;};
public int getWidth() {return width;};
public int getDepth() {return depth;};
public int getHeight() {return height;};
```

3. Rambler's costs

The rambler steps one pixel at a time North, South, East or West (not diagonally). An upward step is more costly:

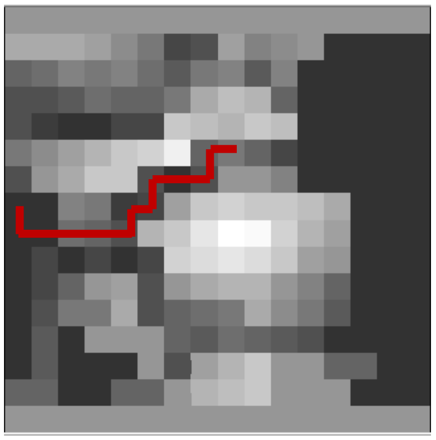
The local cost $\boldsymbol{c(y,x,y',x')}$ of a step from $(\boldsymbol{y},\boldsymbol{x})$ to $(\boldsymbol{y'},\boldsymbol{x'})$ is

$\mathbf{1}$ if $\boldsymbol{h}(\boldsymbol{y'},\boldsymbol{x'}) <= \boldsymbol{h}(\boldsymbol{y},\boldsymbol{x})$ or $\mathbf{1} + |\boldsymbol{h}(\boldsymbol{y'},\boldsymbol{x'}) - \boldsymbol{h}(\boldsymbol{y},\boldsymbol{x})|$ otherwise.

where $\boldsymbol{h}(\boldsymbol{y},\boldsymbol{x})$ is the height in the terrain map at point $(\boldsymbol{y},\boldsymbol{x})$.

The global cost of the walk is the sum of the local costs.

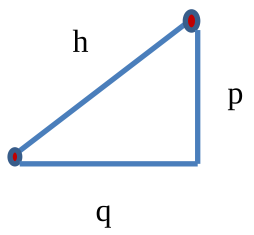
Here is the least cost route in tmc.pgm from the car park at (7,0) to the col at (5,8)



4. What you must do

Using the Java Code provided for COM1005,

1.
   a. Working with the **search3** code and following the procedure in section 2.9 of the notes, implement branch-and-bound search for rambler's problems. You will need to define a class RamblersState and a class RamblersSearch. Look at the corresponding classes for Map Traversal for guidance.
   b. Try out a number of start and end points on the tmc map and assess the efficiency of branch-and-bound in this domain. You may also create other Terrain Maps of your own or make use of **diablo.pgm** in the Ramblers folder which is a terrain map of Mt Diablo in California.[1]
2.
   a. Working from the **search4** code, implement A* search for rambler's problems. For A* you need (under)estimates of the remaining cost to the goal. Consider
      i. the Manhattan Distance between the current pixel and the goal (p+q),



[1] This map is ~ 250*250 points and will take a long time to search. You can reduce the dimensionality by selecting the region you want to search using **crop** in irfanviewer. If you do this, make sure you save the map using **ASCII encoding** (option in pop up window that appears when you do **Save As**)

ii. the Euclidean distance (h),
iii. The Height difference
iv. Combinations of these

You may also devise other ways of obtaining the estimates.

b. Perform experiments to test the hypothesis that A* is more efficient than branch-and-bound and that the efficiency gain is better for more accurate estimates.

You should not have to make any changes to the Java code provided except to control the amount of printout during a search and perhaps to modify what a successful search returns.

5. Mark Scheme

30% of the credit is for 1(a)
20% …………………..1(b)
30% ………………… 2(a)
20% ………………… 2(b)

For your programming, 60% of the credit is for the code itself, 20% testing and 20% for commenting.

6. What to hand in

1. **Commented source code**, ready to run. Give your code for 1(a) and 2(a) separately.
2. **Test results**: each method you write should be separately tested. The tests themselves should cover every logical case and test results should be commented, e.g. sameState: returns true for 2 different instances with identical states, returns false for 2 instances with different states.
3. **Experimental Results** which address the hypothesis in 2(b) above. Consider how best to present results: tables, graphs, bar charts? [2]
Discuss your results – what do they show? What is the best choice of estimates for A*?

7. How to hand in

[2] *You may wish to show solution paths on your terrain maps. There is code to help you do this in the TerrainMap class.. see the README.*

Hand in by MOLE. Put your code, tests and experimentation into a single zip archive & submit that.

8. Deadline

Wednesday of Week 7, 21st March, at midnight.

Late penalties: see

http://www.dcs.shef.ac.uk/intranet/teaching/public/assessment/latehandin.html

Cutoff: 28th March, midnight.