Intelligent Systems Practical 3
Is there life on Mars?

Introduction

This practical asks you to use your knowledge of search algorithms to solve a new kind of problem: planning the route for a robot landing craft on the surface of another planet.

Your mission is to explore the bottom of a Martian crater for signs of life. The area to be explored has already been mapped from orbit and is represented as a grid. Each square in the grid has been classified as accessible or inaccessible.

The toroidal landing craft, Bagel 2, has successfully touched down at the grid square with co-ordinates (4,4), near the centre of the 8×8 grid. It can move north, south, east, or west, to any neighbouring grid square that is accessible. Every time it enters a square it can take a soil sample and conduct an experiment that detects the presence of microorganisms.

However, the landing craft has a limited battery capacity, allowing it to make at most 20 such moves.

The only code you are provided with for this practical is a package called *Mars* with a single class *Planet* that contains a representation of the map of the crater. This class supplies a method *isAccessible(x,y)* that returns a boolean value indicating whether the grid square indexed by x and y is accessible.

- Your first and main task is to find an optimal sequence of moves for the landing craft that maximises the number of different grid squares on which it can perform an experiment. Describe your search strategy, and note how many squares you are able to explore.
- Your second task (which is optional) is to compute the smallest amount of battery life that would allow the landing craft to explore every accessible square in the grid.

You should ensure that the code of your solution is well documented. The deadline for submitting solutions to this practical is week 8.

[If you would like to make this practical harder for next year's students, you are welcome to see if you can find a different map with the same number of accessible squares (or fewer) that requires a larger battery life to explore every accessible square. What is the largest possible battery life that might be required?

I have not found any practical way to tackle these (search) problems. . .]