COMP3411/9414/9814 Artificial Intelligence
Session 1, 2013

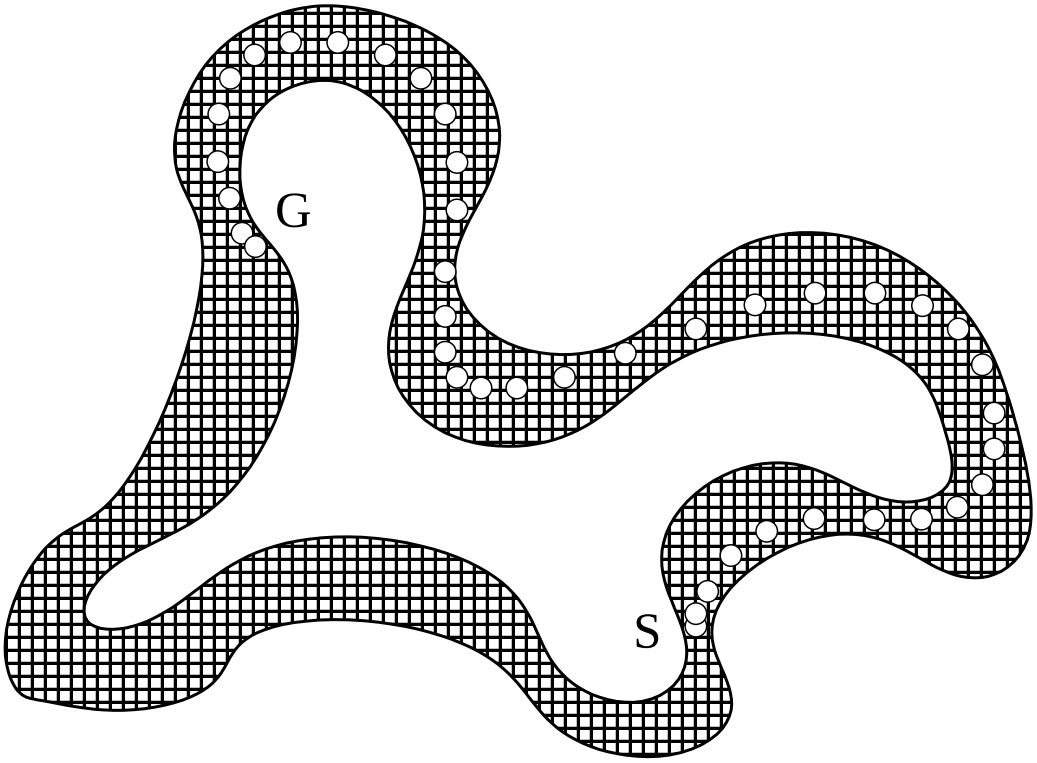
Assignment 2 - Heuristics and Search

Due: Sunday 5 May, 11:59pm
Marks: 8% of final assessment

Question 1 - Graph Paper Grand Prix

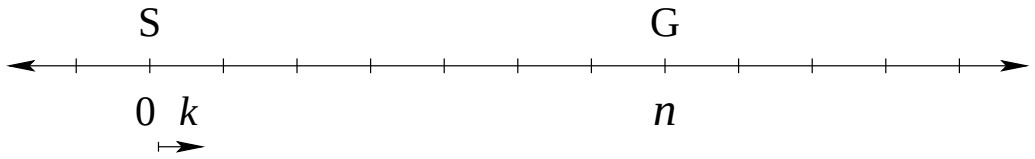
We saw in lectures how the Straight-Line-Distance heuristic can be used to find the shortest-distance path between two points. However, in many robotic applications we wish to minimize not the distance but rather the **time** taken to traverse the path, by speeding up on the straight bits and avoiding sharp corners.

In this question you will be exploring a simplified version of this problem in the form of a game known as Graph Paper Grand Prix (GPGP). [I was first introduced to this game in Year 9 of high school, as a distraction for the last day of class before the summer holiday].



To play GPGP, you first need to draw the outline of a racing track on a sheet of graph paper, and choose a start location S = (r_S, c_S) as well as a Goal location G = (r_G, c_G) where r and c are the row and column. The agent begins at location S, with velocity (0,0). A "state" for the agent consists of a position (r, c) and a velocity (u, v), where r, c, u, v are (positive or negative) integers.

At each time step, the agent has the opportunity to increase or decrease each component of its velocity by one unit, or to keep it the same. In other words, the agent must choose an acceleration vector (a, b) with $a, b \in \{-1, 0, +1\}$. It then updates its velocity from (u, v) to $(u', v') = (u + a, v + b)$, and updates its position – using the *new* velocity – from (r, c) to $(r + u', c + v')$. The aim of the game is to travel as fast as possible, but without crashing into any obstacles or running off the edge of the track, and eventually stop at the Goal with velocity $(0, 0)$.



We first consider a 1-dimensional version of GPGP where the vehicle moves through integer locations on a number line, with no obstacles. Assume the Goal is at location n, and that the agent starts at location 0, with velocity k. We will use $M(n, k)$ to denote the minimum number of time steps required to arrive and stop at the Goal. Clearly $M(-n, -k) = M(n, k)$ so we only need to compute $M(n, k)$ for $k \geq 0$.

(a) Starting with the special case $k = 0$, compute $M(n, 0)$ for $1 \leq n \leq 21$ by writing down the optimal sequence of actions for all n between 1 and 21. For example, if $n = 7$ then the optimal sequence is $[+ + \circ - \circ -]$ so $M(7, 0) = 6$. (When multiple solutions exist, you should pick the one which goes "fast early" i.e. with all the +'s at the beginning.)

(b) Assume $n \geq 0$. By extrapolating patterns in the sequences from part (a), explain why the general formula for $M(n, 0)$ is

$$M(n, 0) = \left\lceil 2\sqrt{n} \right\rceil,$$

where $\lceil z \rceil$ denotes z rounded up to the nearest integer.
Hint: Do not try to use recurrence relations. You should instead use this identity:

$$\left\lceil 2\sqrt{n} \right\rceil = \begin{cases} 2s + 1, & \text{if } \quad s^2 \quad < n \leq s(s + 1) \\ 2s + 2, & \text{if } s(s + 1) < n \leq \ (s + 1)^2 \end{cases}$$

(c) Assuming the result from part (b), show that if $k \geq 0$ and $n \geq \frac{1}{2}k(k-1)$ then

$$M(n, k) = \left\lceil 2\sqrt{n + \tfrac{1}{2}k(k+1)} \right\rceil - k$$

Hint: Consider the path of the agent as part of a larger path.

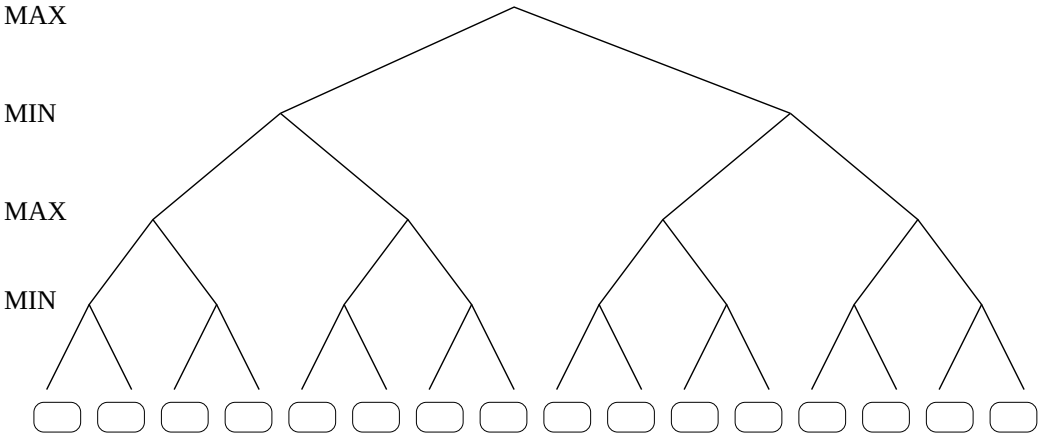
(d) Derive a formula for $M(n, k)$ in the case where $k \geq 0$ and $n < \frac{1}{2}k(k-1)$.

(e) Write down an admissible heuristic for the original 2-dimensional GPGP game in terms of the function $M()$ derived above.
Hint: Your heuristic should be of this form:

$$h\left(r, c, u, v, r_G, c_G\right) = \max(M(..,..), M(..,..))$$

Question 2 - Game Trees and Pruning

(a) Consider a game tree of depth 4, where each internal node has exactly **two** children (shown below). Fill in the leaves of this game tree with all of the values from 0 to 15, in such a way that the alpha-beta algorithm prunes as many nodes as possible. Hint: make sure that, at each branch of the tree, all the leaves in the left subtree are preferable to all the leaves in the right subtree (for the player whose turn it is to move).



(b) Trace through the alpha-beta search algorithm on your tree. How many of the original 16 leaves are evaluated?

(c) Now consider another game tree of depth 4, but where each internal node has exactly **three** children. Assume that the leaves have been assigned in such a way that the alpha-beta algorithm prunes as many nodes as possible. Draw the shape of the pruned tree. How many of the original 81 leaves will be evaluated?

Hint: If you look closely at the pruned tree from part (b) you will see a pattern. Some nodes explore all of their children; other nodes explore only their leftmost child and prune the other children. The path down the extreme left side of the tree is called the line of best play or Principal Variation (PV). Nodes along this path are called PV-nodes. PV-nodes explore all of their children. If we follow a path starting from a PV-node but proceeding through non-PV nodes, we see an alternation between nodes which explore

all of their children, and those which explore only one child. By reproducing this pattern for the tree in part (c), you should be able to draw the shape of the pruned tree (without actually assigning values to the leaves or tracing through the alpha-beta algorithm).

(d) What is the time complexity of alpha-beta search, if the best move is always examined first (at every branch of the tree)? Explain why.

Submission

This assignment must be submitted electronically.

COMP9414/9814 students should submit by typing

```
give cs9414 hw2 ...
```

COMP3411 students should submit by typing

```
give cs3411 hw2 ...
```

The give script will accept `*.pdf *.txt *.doc *.rtf`

If you prefer some other format, let me know.

Late submissions will incur a penalty of 15% per day, applied to the maximum mark.

Group submissions will not be allowed. By all means, discuss the assignment with your fellow students. But you must write (or type) your answers individually. Do NOT copy anyone else's assignment, or send your assignment to any other student.

Good luck!