CITS4211 Mid-semester test 2013

Fifty minutes, answer all four questions, total marks 60

Question 1. (10 marks) Briefly describe the principles, operation, and performance issues of *iterative deepening*.

Illustrate your answer using the tree in Figure 1. Include enough detail to make it clear that you understand how the algorithm works, particularly which nodes are expanded, in what order, and why.

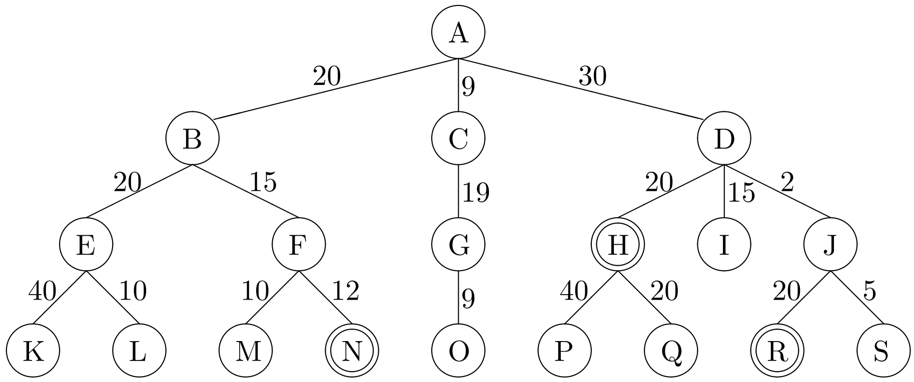

Figure 1: The search tree for Questions 1 and 2. Each arc is labeled with its cost. H, N, and R are the only goal states.

Question 2. (20 marks) Briefly describe the principles, operation, and performance issues of *A**.

Illustrate your answer using the tree in Figure 1 and the heuristic function in Figure 2. Include enough detail to make it clear that you understand how the algorithm works, particularly which nodes are expanded, in what order, and why.

A	42	D	2	G	22	J	12	M	19	P	13	S	13
B	25	E	14	H	0	K	2	N	0	Q	13		
C	34	F	12	I	12	L	10	O	30	R	0		

Figure 2: An admissible heuristic function for the tree in Figure 1.

Question 3. (10 marks) Briefly describe the principles, operation, and performance issues of *minimax*.

Illustrate your answer using the tree in Figure 3. Include all of the detail that you can at each step, to make it clear that you understand how the algorithm works, particularly how the labels of the non-terminals evolve and why, which terminals are ignored and why, and which move is chosen and why. Assume that the children of a node are always examined left-to-right.

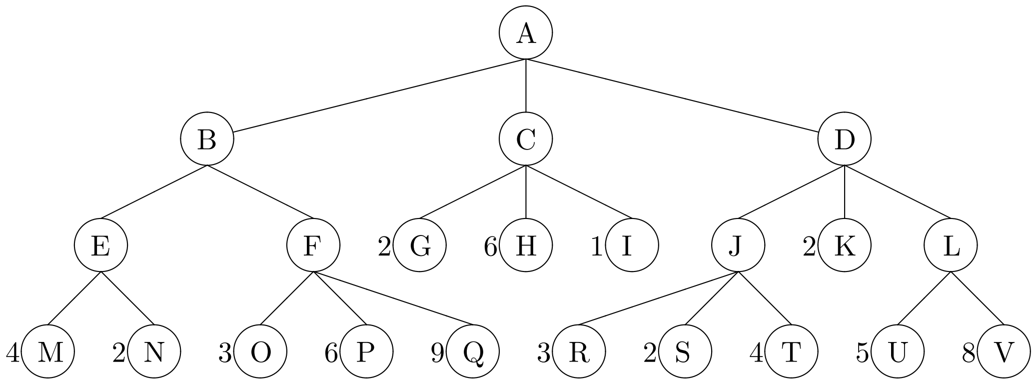

Figure 3: The game tree for Questions 3 and 4. Each terminal node is labeled with its evaluation: higher values represent better positions for the first player.

Question 4. (20 marks) Briefly describe the principles, operation, and performance issues of $\alpha - \beta$ *pruning* as an optimisation to minimax.

Illustrate your answer using the tree in Figure 3. Include all of the detail that you can at each step, to make it clear that you understand how the algorithm works, particularly how the labels of the non-terminals evolve and why, which terminals are ignored and why, and which move is chosen and why. Assume that the children of a node are always examined left-to-right.

1. *Iterative deepening* is a sequence of depth-first, depth-limited searches where the limit starts at 1 and increases by 1 each time. It stops as soon as a goal state is found.

ID has the linear space behaviour of depth-first search, but because of the depth-limit it is complete, and because Node X always precedes Node Y if X is shallower than Y, it is also optimal (it always finds the shallowest goal state). The principal drawback is that some nodes are examined multiple times.

In the example the nodes are expanded in this order:

A; A,B,C,D; A,B,E,F,C,G,D,H

2. *A** maintains a pool of nodes, which starts with just the root. The cost-estimate of each node X is the sum of the cost from the root to X and the heuristic estimate of the cost from X to the goal state. The next node expanded is one which has the lowest cost-estimate. It stops as soon as a goal state is found.

*A** is provably optimal (it always finds the cheapest goal state) as long as its heuristic is admissible, i.e. it never over-estimates the cost from X to the goal state. Variants of *A** (SMA*, IDA*) have been described which also have good space behaviour.

In the example the pool of nodes evolves like this:

A	$0 + 42 = 42$

D	$30 + 2 = 32 = 42$
C	$9 + 34 = 43$
B	$20 + 25 = 45$

C	$9 + 34 = 43$
J	$32 + 12 = 44$
B	$20 + 25 = 45$
H	$50 + 0 = 50$
I	$45 + 12 = 57$

J	32 + 12 = 44
B	20 + 25 = 45
H	50 + 0 = 50
G	28 + 22 = 50
I	45 + 12 = 57

B	20 + 25 = 45
H	50 + 0 = 50
G	28 + 22 = 50
S	37 + 13 = 50
R	52 + 0 = 52
I	45 + 12 = 57

F	35 + 12 = 47
H	50 + 0 = 50
G	28 + 22 = 50
S	37 + 13 = 50
R	52 + 0 = 52
E	40 + 14 = 54
I	45 + 12 = 57

N	47 + 0 = 47
H	50 + 0 = 50
G	28 + 22 = 50
S	37 + 13 = 50
R	52 + 0 = 52
E	40 + 14 = 54
I	45 + 12 = 57
M	45 + 19 = 64

3. *minimax* is an algorithm for making decisions in deterministic, perfect-information, two-player, zero-sum games. The game tree represents all possible sequences of moves. Each leaf node of the tree is assessed using an evaluation function, then upwards from the leaves to the root at each of (the first player) MAX's choice points s/he chooses the move with the highest value, and at each of (the second player) MIN's choice points s/he chooses the move with the lowest value. minimax is complete and optimal for finite trees against optimal opponents. All nodes are inspected.

In the example the labels of the non-terminals evolve like this:

- $M = 4, N = 2 \to E = 4$
- $O = 3, P = 6, Q = 9 \to F = 9 \to B = 4$
- $G = 2, H = 6, I = 1 \to C = 1$
- $R = 3, S = 2, T = 4 \to J = 4$
- $K = 2, U = 5, V = 8 \to L = 8 \to D = 2 \to A = 4$

And MAX chooses B.

4. $\alpha - \beta$ *pruning* implements *minimax*, but it ignores (prunes) nodes when it has enough information to infer that inspecting the node would not change the value assigned to one of the node's ancestors.

$\alpha - \beta$ obviously in general inspects fewer nodes than *minimax*: the saving will be bigger if it inspects relatively poor nodes as early as possible in the process.

In the example the labels of the non-terminals evolve like this:

- $M = 4 \to E \geq 4$
- $N = 2 \to E = 4 \to B \leq 4$
- $O = 3 \to F \geq 3$
- $P = 6 \to F \geq 6 \to B = 4 \to Q$ is irrelevant and $A \geq 4$
- $G = 2 \to C \leq 2 \to H, I$ are irrelevant
- $R = 3 \to J \geq 3$
- $S = 2 \to$ nothing
- $T = 4 \to J = 4 \to D \leq 4 \to A = 4$ and K, L, U, V are irrelevant

And MAX chooses B.