CMPUT 396	3 hr	closedbook	6 pages, 7 marks/page	page 1

id#

first names

1. [3 marks] For each person or program, give the label of its description.

last name

 Aja Huang
 Cho Chikun
 David Silver
 Demis Hassabis

 Fan Hui
 Geoff Hinton
 Lee Sedol
 Michael Redmond

(a) former European go champion who lost to a strong computer program in a 5-game match in 2015 (b) CEO of a leading AI research company (c) lead programmer of AlphaGo (d) former #1 go player who lost to a strong computer program in a 5-game match in 2016 (e) 9dan professional go player and commentator (f) neural net expert who wrote a paper on image classification (g) first author of Nature paper on AlphaGo (h) former #1 go player who defeated a strong computer program in a 3-game match in 2016 .

2. [4 marks] Fill in the blanks, and circle correct answers.

AlphaGo integrates neural net calls into ________ search, overcoming the slowness of net calls by (circle all that apply) a) building shallow nets that reply quicker than the initial deeper versions b) handling net calls with GPUs c) having the master algorithm continue to operate while net calls are executing d) having each net call distributed over parallel processes.

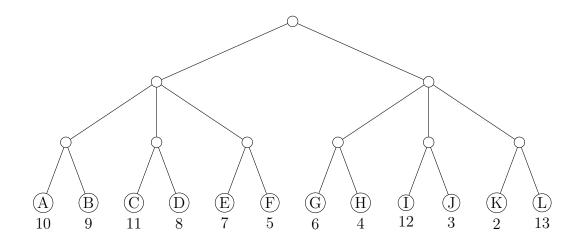
AlphaGo child-selection uses (circle all that apply) a) a deep policy net b) a deep value net c) a shallow policy net d) a shallow value net e) simulations.

In AlphaGo, once a leaf is reached, using a **(circle all that apply)** cpu / gpu , a call is made on a **(circle all that apply)** a) deep policy net b) deep value net c) shallow policy net d) shallow value net e) simulation net .

Also, at the leaf, a simulation is performed on a **(circle all that apply)** gpu / cpu . Then, using a fractional weighting of ______ for the call and ______ (these 2 fractions sum to 1.0) for the simulation, the leaf score is backed up the search tree.

last name		first names		$\mathrm{id}\#$
CMPUT 396	3 hr	closedbook	6 pages, 7 marks/page	page 2

[4 marks] This is a minimax tree. The root player is max. Each leaf label is the root player's score for that leaf. i) On the diagram, beside each non-leaf node, write the root player's minimax value for that node. ii) Assume that minimax values are found by (recursive) alphabeta search, with children of a node considered starting from the left. For this search, on the diagram, draw 2 short lines through each edge that is pruned, and draw a box around each leaf node that is examined.



a b cFor this tic-tac-toe position with x to move, the minimax value is
(circle one) x-win draw o-win. A best move for x is _____
and a best reply for o is _____. A simple minimax search would
consider about this many states: (circle one) 3 30 300 3000
30000 300000 7! = 5040 8! = 40320 9! = 362880.

for . . rough work

last name		first names		$\mathrm{id}\#$		
CMPUT 396	3 hr	closedbook	6 pages, 7 marks/page	page 3		

5. [4 marks] In MCTS, for a child with w wins and v visits, the function f(w, v) = (w + t)/(v + 2t) is used to measure win rate instead of g(w, v) = w/v because (circle all that apply) a) f allows the true value to be estimated more quickly b) f returns a value when v is 0, so never divides by 0 c) f increases the statistical significance of the simulations d) f allows quicker recovery from initial unlucky simulations.

The MCTS UCB1 formula balances the exploitation of a ______ search with the ______ of children which have received fewer ______ than their siblings. For each child j, the formula is (circle one) a) $f(w_j, v_j) + c\sqrt{\ln(v_1 + \ldots v_t)/v_j}$ b) $f(w, v) + c\sqrt{(v_1 + \ldots v_t)/v_j}$ c) $f(w_j, v_j) + c\sqrt{v_j/(v_1 + \ldots v_t)}$ d) $f(w, v) + c\sqrt{v_j/\ln(v_1 + \ldots v_t)}$. MCTS can be improved by adding patterns to simulations: eg. in Go, after each simulation move, if a move creates a match with a local (ie. around that move) pattern, then (circle one) a) a random move is performed b) the reply move for that pattern is played c) the appropriate player is designated the winner d) the leaf node has its RAVE count increased by 1. Eg. in Go, if a white simulation move is as shown in this local 2×2 pattern

Eg. in Go, if a white simulation move is as shown in this local 2×2 pattern sequence, then what happens next is ______



6. [3 marks] Before 2000, the strongest Go program was as strong as a human with rank (circle one)
a) 5 dan b) 9 dan c) 15 kyu d) 30 kyu. MCTS was first used in Go programs around the year ______. Later Clark and Storkey used records from ______ professional games to build a deep _______ neural net that predicts the most popular move correctly with probability about ______. Eg. their net predicts that the first move on the empty ______ x _____ board will be at the (circle one) a) 4 4 point b) 5 5 point c) 6 6 point d) 7 7 point. Later the company _______ (owned by Google) wrote AlphaGo, which is about as strong a human with rank (circle one) a) 5 dan b) 9 dan c) 15 kyu d) 30 kyu.

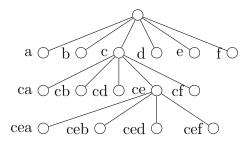
last name		first names		$\mathrm{id}\#$	
CMPUT 396	3 hr	closedbook	6 pages, 7 marks/page	page 4	

7. [4 marks] The Tromp-Taylor rules use _______ superko: a move cannot recreate any previous position. Eg. assume from this 1×5 Go state black moves to cell 5, resulting in ______ in move white cannot move to cell ______ because of superko.

For each n in the table, give the first-player minimax value for $1 \times n$ Go.

n	1	2	3	4	5	6
value						

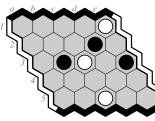
8. [3 marks] For a position in a 2-player game with players x, o and player-to-move x, here is a mcts tree at some point in execution. Node labels show the associated move. Now a simulation occurs at the leaf node whose path from the root is -c-e-b, playout -f-d-a, result x win. For this extended playout, the moves made by x were ______ and by o were ______. In the table below, give the change (leave it blank if no change) to each node's wins, visits, rave-wins, rave-visits that happens during backup. Column a will be all blank.



	a	b	с	d	е	f	ca	$^{\rm cb}$	cd	ce	cf	cea	ceb	ced	cef
W															
V															
ravew															
ravev															

last name		first names	s id	$\mathrm{id}\#$		
CMPUT 396	3 hr	closedbook	6 pages, 7 marks/page	page 5		
[3 marks] For	this Hex posit	ion, after black plays a	t b4, black has a winning virtual			
connection us	ing cells {a5,b	$\{a1, a2, a3, b1, b2\}$	2,c1,c2,d1}. Similarly, after black			
plays at c2, bl	ack has a winn	ing virtual connection	using cells $\{c1,d1\}$ and $\{a4,a5,b4\}$			

and {e2,d3} and {e4,d5,e5}. So, for this position with white to play, white must play at one of the cells in { _______}}, otherwise black can win. For this position with white to play, the set of all winning white moves is { _______}}. So far, the largest Hex boardsize on which winning opening moves have been found is _______.



10. [2 marks] For the nim state with piles 10 9 8 5, list all winning moves below (if there is no winning move, leave the blank empty). On the side, show your work.

From the 10 pile, remove ______ stones.

From the 9 pile, remove ______ stones.

From the 8 pile, remove ______ stones.

From the 5 pile, remove ______ stones.

5 2 3The number of inversions of this sliding tile puzzle is ______ (an11. [2 marks] 1 8 4______ number) and the number of columns is an odd number, so7 6 _this puzzle (circle one) is / is not solvable.

Consider Python implementations of these algorithms that solve 5x5 (and smaller) sliding tile puzzles: the A* algorithm described in class, and a special-purpose (SP) algorithm using the method from the youtube video discussed in class. (circle all that apply)

a) A* with the Manhattan heuristic is usually faster than A* with the misplaced tiles heuristic.

b) A* with the Manhattan heuristic is usually faster than SP, because it finds a shortest path through the state space.

c) A* with the Manhattan heuristic is usually slower than SP, because SP does not search the whole state space.

d) The runtime for SP on 5x5 puzzles is about 1.25 time the runtime for SP on 4x4 puzzles.

first names

CMPUT 3963 hrclosedbook6 pages, 7 marks/pagepage 6

- 12. [1 marks] Recall: in Go a group of stones is unconditionally safe if the opponent cannot kill the group, even if the player always passes. The simplest kind of unconditionally safe group is one that has at least _______ ____.
- 13. [6 marks] Consider these games: hex on an 6×6 board, go on a 1×18 board, and tic-tac-toe on an 6×6 board (where to win, you need 4 in a row). For each game you want to write a computer solver (so, an agent that finds a move with best minimax score).

For 6x6 hex, (circle all that apply)

a) the game can be solved in a reasonable amount of time using only alphabeta search, since the game tree has only $36! \approx 4 \times 10^{41}$ nodes

b) implementing a transposition table is not difficult, especially since there are no draws

c) the game can be solved in a reasonable amount of time using only alphabeta search, a transposition table, and symmetry pruning, since the solving dag then has about 10^{30} states

d) the game can be solved in a reasonable amount of time using alphabeta search, a transposition table, and pruning with symmetry, mustplay, and inferior cells, since the solving dag then has about 10^{15} states

For 1x18 go, (circle all that apply)

a) the game can be solved in a reasonable amount of time using only alphabeta search since the game tree has only $18! \approx 6 \times 10^{15}$ nodes

b) implementing a transposition table is not difficult, since the winner scores at most 18 points

c) the game can be solved in a reasonable amount of time using only alphabeta search and a transposition table since the solving dag then has about 10^{12} states

d) the solver can be improved by recognizing commonly occuring unconditionally safe groups .

For 6x6 tic-tac-toe, (circle all that apply)

a) the game can be solved in a reasonable amount of time using only alphabeta search since the game tree has only 36! nodes

b) implementing a transposition table is not difficult, as there are only 3 possible outcome values,

c) the game can be solved in a reasonable amount of time using only alphabeta search and a transposition table, since the game is likely to end in a draw,

d) the solver can be improved using mustplay pruning (ie. play here or lose) .