CPSC 474/574 - Spring 2018 - Exam #2

Write your name and NetID on this cover page and only on this cover page. Please indicate clearly your final answers to each question. Do not write within $\frac{1}{2}$ inch of the edges of the pages.

Problem 1 (12 points): Consider a two-player game played with n piles of stones each initially containing k stones. On each turn, the current player may take one stone from each remaining pile, or divide any pile containing at least 2 stones into 2 or 3 non-empty piles. For example if the sizes of the piles are $[3, 2]$ then the possible moves are to $[2, 1]$ (by taking one stone from each pile), to $[3, 1, 1]$ (by dividing the pile of 2 into two piles of 1), to $[2, 1, 1, 1]$ (by dividing the pile of 3 into three piles of 1), and to $[2, 2, 1]$ (by dividing the pile of 3 into a pile of 1 and a pile of 2). The player who makes the last move wins.

Find the Grundy numbers of every position reachable starting with one pile of 4 stones.

4

3 1

2 2

2 1 1

1 1 1 1

3

2 1

1 1 1

2

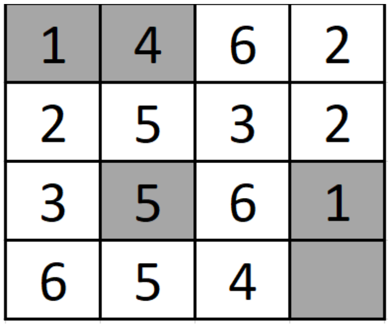
1 1

1

0

Problem 2 (15 points): Consider a solitaire game played on a h-row, w-column grid of positive numbers. You start with a marker on the bottom right corner of the grid (location $(h-1, w-1)$). On each turn you can choose to move left or up, and your marker moves to a position in that direction chosen uniformly randomly and the number in the resulting position is added to your score. If your marker is already in the top row or left column then you may not choose up or left respectively. The game ends when the marker is in the top left corner (position $(0,0)$).

For example, in the pictured game the player chose up, left, up, and left and the randomly selected moves were 1, 2, 2, and 1 spaces respectively. The total score was 11 (note that it does not matter what is in the lower right corner since that value is never added to the score).



1	4	6	2
2	5	3	2
3	5	6	1
6	5	4	▨

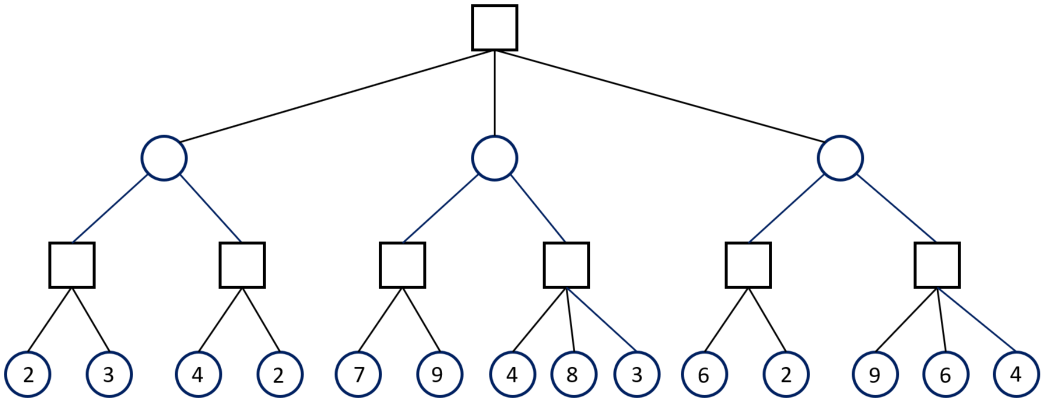
Let $E(r, c)$ be the expected future winnings when following the optimal strategy starting at position (r, c). Write a recurrence (including the base cases) for $E(r, c)$. Assume that $A[r, c]$ holds the value of the number at row r, column c.

Problem 3 (12 points): Illustrate the operation of Scout on the given game tree by showing

(a) any pruning that occurs during the initial calls to the children of the root (not any pruning that happens during any re-searches)

(b) the null windows that are passed initially to the second and third children of the root

(c) whether or not a re-search is required after the initial null-window calls to the second and third children of the root, and if so, the full window then passed to the child to start the re-search.



squares are max nodes; circles are min nodes

Problem 4: (12 points) Consider the constant sum game with the following payoff matrix.

$$\begin{pmatrix} 6 & 3 \\ 2 & 4 \end{pmatrix}$$

For each of the following pairs of mixed strategies, determine if it is a saddle point. If it is, then show that it is and find its value; if it is not then find player I's best response to the given strategy for player II (find the best response to Y_i).

(a) $X_1 = (\frac{1}{2}\ \frac{1}{2}), Y_1 = (\frac{1}{3}\ \frac{2}{3})$

(b) $X_2 = (\frac{2}{5}\ \frac{3}{5}), Y_2 = (\frac{1}{5}\ \frac{4}{5})$

Problem 5: (10 points) Consider a multi-armed bandit with three arms: one that pays 1 or 6, each with probability $\frac{1}{2}$; one that pays 0 or 8, each with probability $\frac{1}{2}$; and one that pays 2 or 3, each with probability $\frac{1}{2}$.

Consider a policy that plays each arm once, eliminates the arm with the lowest payoff during those plays, and then alternates playing the remaining two arms. Compute the exact value of the expected value of the limit of the average regret for this policy on this machine.

Problem 6: (16 points) Describe one way of incorporating domain knowledge in Monte Carlo Tree Search.

Problem 7: (8 points) The following code implements part of the tree policy for Monte Carlo Tree Search for a one-player game. $r[pos]$ is the average observed reward for a position, $n[pos]$ is the number of times a position has been reached by the search, and $moves(pos)$ returns a list of the positions reachable in one move from position pos (assume that the game tree is in fact a tree, so there is at most one path from the root to any given position).

```
BEST-CHILD(p)
  max = -infinity
  move = null

  for c in moves(p)

    exploit = r[p]

    explore = sqrt(2 * log(n[p]) / n[c])

    ucb = exploit + explore

    if ucb > max
      max = ucb
      move = c

  return move
```

Modify the above code to work for a two-player, zero-sum combinatorial game, assuming that

- for a given position p, the next player to move at some positions in $moves(p)$ could be the same as the player to move at position p and could be different for other positions in $moves(p)$, and
- $r[p]$ is the observed reward at position p from the point of view of the next player at p: positive if that player is likely to win and negative otherwise.

You may assume that there is a function $next(p)$ that returns the index of the player making the next move at position p.

Problem 8: (15 points) Describe the inputs you might use for an artificial neural network that, for a position in 2-player Yahtzee, estimates the probability that the first player wins (a value network for 2-player Yahtzee).