- Q1

1. Consider the traveling salesman problem (TSP) example, with only a few nodes (4-5).
   Then, explain how the following algorithms work and how they can be applied to solve the TSP:
   A. Hillclimbing search
   B. Simulated annealing
   Clearly show the basic math and the steps involved, and how to apply them to TSP. [10 points]

V={c1, c2, ⋯, ci, ⋯, cn}, i = 1,2, ⋯, n, is the set of all cities. ci represents the i-th city, and n is the number of cities;
C = {crs: r,s∈V} is the cost metric of the connection between all cities (usually the distance between cities);
A TSP problem can be expressed as: Solve the traversal graph G = (V, E, C), all nodes once and return to the starting node, making the path cost of connecting these nodes lowest.
A: heuristic cost function: the lower total distance travelled with successor state
Successor state: a set of the next unvisited nodes
1. choose an initial state, and the solution Xbest = x0.
2. choose a successor state according to the heuristic cost function.
3. if distance(Xnow) is lower than Xbest, then let Xbest = Xnow, back to the second step.
4. if there is no successor state then stop.
B:
1. Set the initial temperature and initialize an initial solution randomly.
2. Begin to enter the outer circulatory body.
2.a Enter the internal circulation body.
2.a.a In the existing solution, the city visit sequence, randomly exchange the order of one or more groups of cities to generate a new solution.
2.a.b Record the optimal solution of the inner loop.
2.a.c exit the inner loop (I used the number of iterations k for the termination condition)
2.b According to the metropolis criterion, judge whether to accept the new solution based on the total distance of the new solution and the total distance of the new solution and the current distance t, that is, replace the old solution with the new solution;
2.c The temperature drops
3. if he temperature drops to a certain degree, exit the outer loop

- Q2

4.2 Exercise 3.16 considers the problem of building railway tracks under the assumption that pieces fit exactly with no slack. Now consider the real problem, in which pieces don't fit exactly but allow for up to 10 degrees of rotation to either side of the "proper" alignment. Explain how to formulate the problem so it could be solved by simulated annealing.

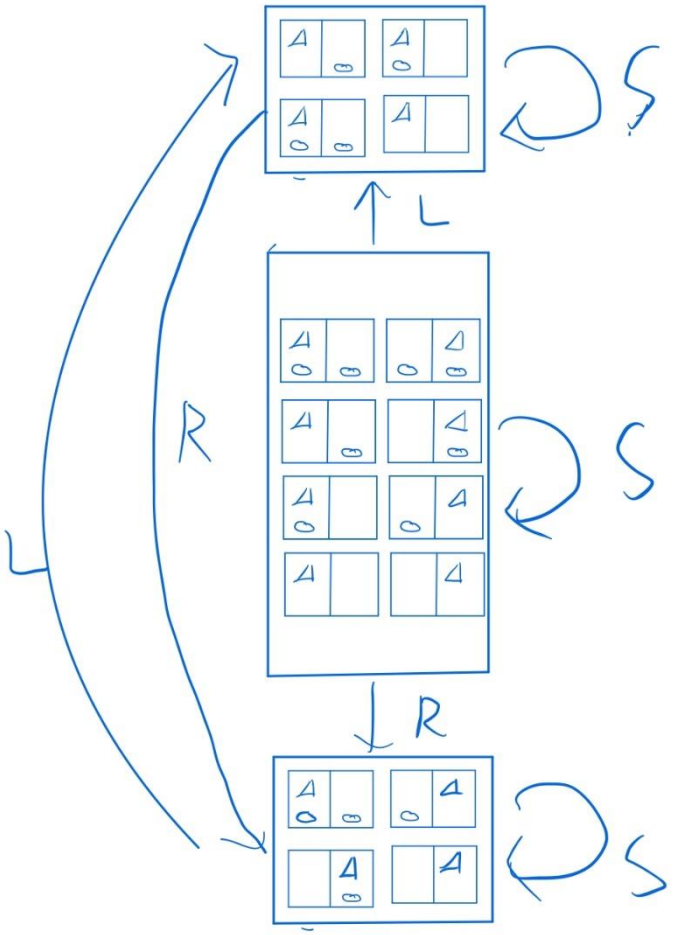
STATE: a set of linked pieces and the angles are in the range of [-10,10], and a set of unlinked pieces.
EVALUATION FUNCTION: how many pieces are used, how many loose ends there are and the degree of overlap.

The tricky part the set of allowed moves. Any unlinked pieces could be linked to others or an open peg at any angle, and moves to join a peg and hole on linked pieces are also problematic. And changing on one angle will lead to the changes on another pieces. So there may be no minimal solution for a fixed angle change.

- Q3

4.10 Consider the sensorless version of the erratic vacuum world. Draw the belief-state space reachable from the initial belief state $\{1, 2, 3, 4, 5, 6, 7, 8\}$, and explain why the problem is unsolvable.

No path could resulted in the state satisfy the goal. So the problem is unsolvable



- Q4

4.12 Suppose that an agent is in a 3×3 maze environment like the one shown in Figure 4.19. The agent knows that its initial location is (1,1), that the goal is at (3,3), and that the actions *Up*, *Down*, *Left*, *Right* have their usual effects unless blocked by a wall. The agent does *not* know where the internal walls are. In any given state, the agent perceives the set of legal actions; it can also tell whether the state is one it has visited before.

a. Explain how this online search problem can be viewed as an offline search in belief-state space, where the initial belief state includes all possible environment configurations. How large is the initial belief state? How large is the space of belief states?

In belief-state space includes all possible environment configurations, online search is equal to offline search. Each action could lead to many successor states because agent could see the percept after the action. After the action, each state is replaced by the set of successor state, and then the successor states are not consistent with the percept will be removed.

AND-OR could be used to solve this problem. There are 1024 Initial states and 2212 possible belief states.

- Q5

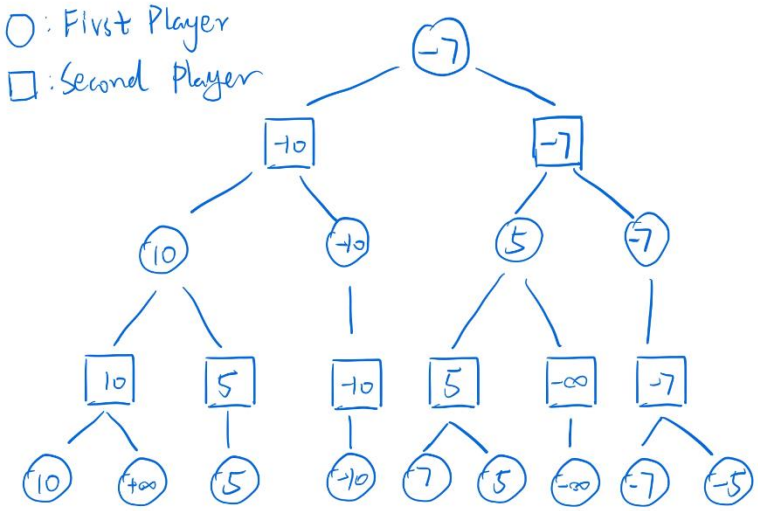
5. MiniMax Algorithm **[10 points]**

In the context of a simple board game of your choice, answer the following questions:

A. How were the game states represented as a game tree? Explain with pictures.

Starting from the current state or initial state, according to the actions that can be taken and the states that can be reached, draw the child nodes, and then draw the child nodes for the child nodes. State until the end of the game.

Suppose there is a game where only two players participate, and two rounds will be the winner. Each player has 1 or 2 actions to choose from in their respective rounds. First player goes first. Each link represents the action that can be taken in the current state. Because it is assumed that the game has only two rounds, when the game progresses to the layer marked 4 in the figure, the outcome of the game is already known.



Suppose you are in the layer 3, which is the player 2's round. player 2 will definitely choose the action that makes player 1's score the least among all the available actions. Therefore, fill in the smaller utility value of the child node into the third layer, which indicates the maximum utility that player 1 can achieve when the game progresses to the corresponding state. Then back to the second layer, this layer is player 1's round, player 1 knows its own utility value in various states of the third layer, player 1 will of course choose the action that maximizes its own utility. Therefore, select the child node with the greatest utility to fill in the second layer. Then go back to the first level, which is the player 2's round, the same as the third level. Finally back to the 0th level, the player 1's round, the same as the 2nd level. player 1 started backtracking from the end of the game, knowing that the maximum utility that he can achieve is -7.

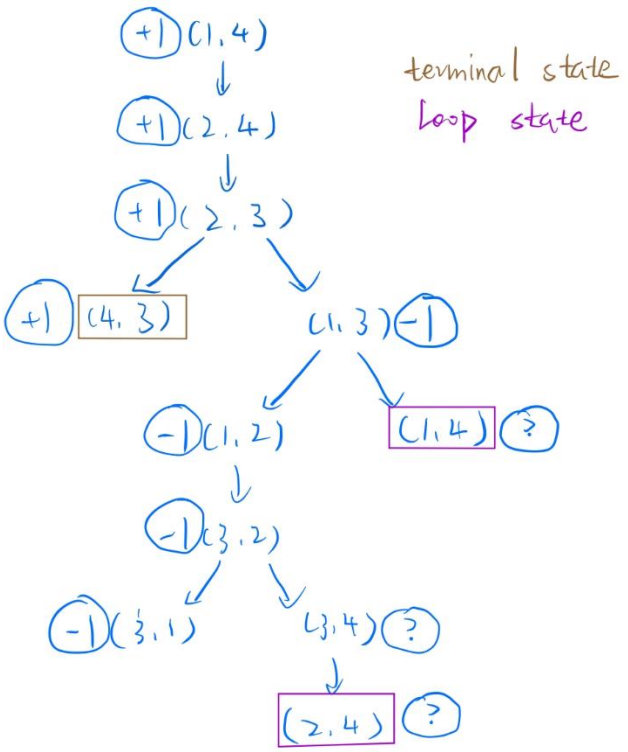
- Q6A

5.8 Consider the two-player game described in Figure 5.17.

a. Draw the complete game tree, using the following conventions:

- Write each state as (s_A, s_B), where s_A and s_B denote the token locations.
- Put each terminal state in a square box and write its game value in a circle.
- Put *loop states* (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a "?" in a circle.

b. Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the "?" values and why.



(a)

(b) Min(-1,?) is -1 and max(+1,?) is +1, and all successors are ? will return ?. Assuming agent will choose to win rather than other choices.

- Q6B

5.12 Describe how the minimax and alpha–beta algorithms change for two-player, non-zero-sum games in which each player has a distinct utility function and both utility functions are known to both players. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha–beta? What if the player's utility functions on any state differ by at most a constant k, making the game almost cooperative?

There is no change between two-player and non-zero-num games.
EVALUATION FUNCTION: a vector of values for each player. Which vector has the highest value will be selected for the backup step.
In general games, alpha-beta pruning will not be used because the original leaf node will be the optimal conditions.

- Q7

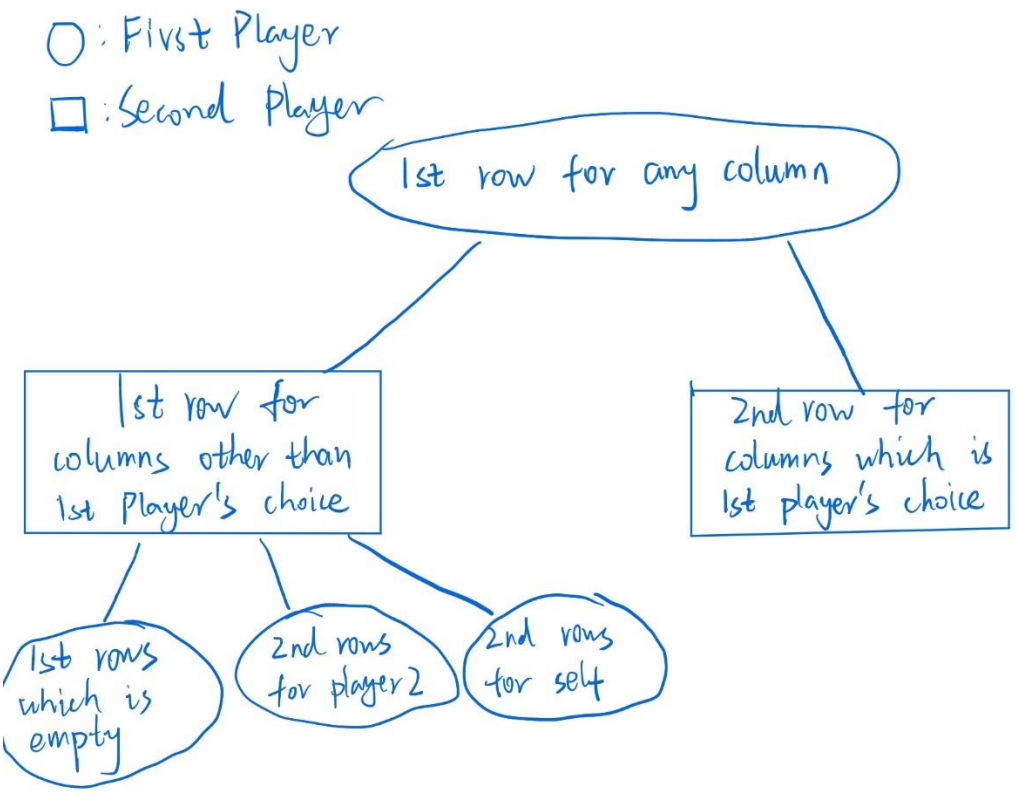
7. Study how the Connect Four game works. In connect-four, players alternately drop discs into the columns of a vertically positioned grid; the first player to get four of her discs in a row wins (if the board fills up before this happens, it's a draw). [25 points]

For this game, explain the below:

A. How is a Game Tree constructed (a small sample of states).
B. Formal definition of this game as an Adversarial Search problem.
C. Explain with a diagram and pseudocode how Mini-Max can be used.
D. How does Alpha Beta pruning work for the game, what does it improve?
E. How would you apply a Cutting Off Search strategy? What would be the heuristic?

1

A. The game Tress is constructed as follows



B:

State set: S (every empty in the grid)

Players: P={1,2}

Action set: A Transition Function: put a disc in the vertical grid.

Terminal Test: There are four same discs in a row

Terminal Utilities: The sum of steps

C:

D:

Alpha-beta pruning work for this game by deleting the branch which would lead to the wins of computer in two or three steps. It could avoid large calculation.

E:

We set a fixed depth, then when depth is larger than fixed depth, cutoff-test would return true. The depth could be the distance to the top of the grid.