8-puzzle:
Given a sliding-block puzzle with 8 tiles, find a sequence of moves that will produce a goal configuration.

Search: find a sequence of actions that can produce a desired outcome

- Single-player: non-adversarial
- Multiple players: adversarial

\[
\begin{array}{ccc}
1 & 2 & 3 \\
8 & 4 & 7 \\
6 & 5 & \\
\end{array}
\]

\[
\begin{array}{ccc}
2 & 6 & \\
3 & 1 & 4 \\
7 & 6 & 8 \\
\end{array}
\]
Probabilistic 1+2-person games: adversarial

because player does not have total control

Maze Traversal:

Initial Situation
Possible Moves (actions)
Goal Condition

Missionaries & Cannibals

- 3 M's + 3 C's on one bank of river
- 2 person boat
- if cannibals outnumber missionaries on one bank, they will kill them
Goal - safely transport all 6 people across river

Initial State: (MMM CCC B)(C)

Goal: () (MMM CCC B)
Backtrack( StateList ) // returns path to goal, or "Failure"

    State = first( StateList )
    if goal(State), return StateList
    if deadEnd(State), return "Failure"
    if State is a member of rest( StateList ), return "Failure"

    Moves = all possible moves from State
    for each m in Moves:
        nextState = applyMove(m,State)
        newStateList = nextState + StateList  // add nextState to front of list
        path = Backtrack( newStateList )  // add nextState to front of list
        if path != "Failure", return path

    return "Failure"  // you only get here if all moves resulted in failure
Depth-first: always choose the most recent new node and expand it.

Breadth-first: always choose nodes in "First come, first served" order.
Best-first search:

Use a "ranking" to decide which node to expand next ("heuristic")

For example—let $f(\text{Board}) = \# \text{tiles out of place}$

\[
\begin{array}{c}
\text{state} \\
\begin{array}{ccc}
2 & 5 & 3 \\
1 & 6 & 4 \\
7 & 8 & 5 \\
\end{array} \\
\hline
f(\ ) = 4 \\
\end{array}
\quad
\begin{array}{c}
\text{goal} \\
\begin{array}{ccc}
1 & 2 & 3 \\
8 & 4 & \text{ } \\
7 & 6 & 5 \\
\end{array} \\
\end{array}
\]
\[ F(\text{Board}) = \text{depth in tree} + \# \text{tiles out of place} \]
Programming w/ Backtrack to solve certain AI problems

- how to represent a state

Missionaries and Cannibals:

- list: left: M M C C B right: M C
- 2 2 L 1 1
  - M on left, C on left, boat pos 1, M on right, etc.

- how do you represent the knowledge needed for this problem
how to represent moves

precondition: must be true for the action to be performed
action: transforms the state

All possible actions:

(11R)(20R)(02R)(01R)(10R); move Mr. Mission and n Cannibals
(11L)(20L)(02L)(01L)(10L) in direction d

precondition: if position of boat = L, then only moves of form (m n R) are possible
function All Possible Moves (state):
    for each m in All Possible Actions
        if precondition(m, state)
            PossibleMoves.append(m)
    return PossibleMoves

function goal (state):
    // for M+C problem -
    is goal == "0 0 33 Boat"
Apply Move:

// For m & c problem:

if \((m, n, B, x, y) = \text{state}\) then

\[ \text{move} = (a, b, \text{dir}) \]

# on left # on right
boat # to move
po6

new state = either

\((m + a, n + b, \text{dir}, x - a, y - b)\)

or

\((m - a, n - b, \text{dir}, x + a, y + b)\)
Encoding for Moves:

\[
\text{ALL MOVES} = \begin{pmatrix} (20)(10)(11)(01)(02) \\ (-20)(-10)(-11)(0-1)(0-2) \end{pmatrix}
\]

A move \((x, y)\) means moving \(x\) missionaries and \(y\) cannibals from left bank to right bank.

\[
\text{STATE} = (3, 3, L)
\]

\[
\text{new STATE} = (3+x, 3+y, R)
\]

Write a function that returns the new state after applying move.
**Initial State** = (3, 3)

**Allowed** = (2, 0)(1, 0)(1, 1)(0, 1)(0, 2)
(2, 0) (1, 0) (1, -1) (0, -1) (0, -2)

**Precondition** (state, move) = TRUE if:

- $0 \leq m+x \leq 2$ \& $0 \leq c+y \leq 2$
- position of boat is appropriate:
  - $pos = \text{RIGHT} \Rightarrow x, y \geq 0$
  - $pos = \text{LEFT} \Rightarrow x, y \leq 0$

**Apply Move** (state, move)

**Returns** (m+x, c+y, other pos)

**Goal**: state = (0, 0)

**Deadend**

- Cannibals Kill Missionaries
  - (Fate at 11)

- $(mL > 0 \ \text{LL} \ cL > mL) \|$
- $(mR > 0 \ \text{LL} \ cR > mR)$

- given $(m, c) = \text{state}$
  - $m$: # missionaries on left = mL
  - $c$: # cannibals on left = cL
  - $3-m$: # missionaries on right = mR
  - $3-c$: # cannibals on right = cR