• **State Space**
  - States: initial, goal, state description
  - State transition rules/operators/actions, and costs associate with operations
  - State space as directed graph (nodes, arcs, parents/children)
  - Node generation and node expansion: open/closed nodes, open/closed lists.
  - **Solution, solution path and its cost.**
  - Be able to represent simple problem-solving as state space search

• **Uninformed (blind) Search Methods**
  - Breadth-first.
  - Depth-first, Depth-limited (plus back-tracking).
  - **IDDF**: Iterative-deepening depth-first. (motivation, advantage over BF and DF methods.)
  - Uniform-cost search.
  - Bi-directional search (advantage and applicability)
  - **Algorithm, time and space complexities, optimality and completeness** of each of these search methods

• **Informed Search methods**
  - Evaluation function f(n),
  - **Heuristic estimate function** h(n)
    - what does h(n) estimate
    - admissible h(n), null h(n), perfect h*(n), more informed h(n)
    - idea of automatic generation of h functions
  - Best first search: open list is organized according to f(n)
  - Algorithm A and A*
    - f(n) = g(n) + h(n): what does each of the terms stand?
    - algorithm (maintaining open/closed lists, delayed termination test; node expansion and generation, handling duplicate nodes, back pointers);
    - difference between algorithms A and A*
    - time and space complexity, completeness and optimality of A*
    - **be able to apply A* to simple problems**.
  - Greedy search and hill-climbing (algorithms; time and space complexity, completeness and optimality)

• **Game -Tree Search**
  - Perfect 2-player games
  - Game tree (Max and Min nodes; terminal and leave nodes)
  - **What to search for** (one move for Max)
  - Heuristic evaluation function f(n) (merit of a board configuration)
  - **Minimax rule for game tree search**
  - **Alpha-beta pruning**, its time and space complexities.
  - Difference between general state space search and game tree search
- Be able to apply Minimax rule and alpha-beta pruning to simple problems.

- **Propositional Logic (PL)**
  - Syntax
    o Propositions
    o Symbols (T, F, proposition symbols)
    o Connectives
    o Definition of PL sentences
  - Semantics
    o **Interpretation** (an assignment of truth values to all prepositional symbols); models
    o Truth tables for logical connectives
    o Valid (tautology), satisfiable, and inconsistent (contradiction) sentences
    o **Logical consequence or theorem** \( (S \models X) \)
  - Equivalence laws
    o \( P \equiv Q \) iff they have the same truth tables
    o \( P \Rightarrow Q \equiv \neg P \lor Q \); distribution /associative/communicative laws, De Morgan's laws
  - Deductive rules
    o Derivation using inference rules: \( S \vdash X \)
    o Modus Ponens, Modus Tollens, Chaining, And Introduction, And Elimination, etc.
    o **Resolution rule (and CNF)**
  - Deductive inference
    o Using truth table \( (S \models X \text{ if } S \vdash X \text{ is valid}) \)
    o Proof procedure (using inference rules)
    o **Sound** inference rules and proof procedures (if \( S \models X \text{ then } S \vdash X \))
    o **Complete** proof procedures (if \( S \models X \text{ then } S \vdash X \)). (exponential time complexity)

- **First Order Logic (FOL)**
  - Syntax
    o Terms (constants, variables, **functions of terms**)
    o Predicates (special functions, ground predicates), atoms and literals
    o Logical connectives
    o **Quantifiers** (universal and existential), their scopes, De Morgan's law with quantifiers
    o Definitions of FOL sentences and well-formed formulas (wffs)
  - Semantics
    o Interpretation (constants, functions, and predicates) and models
    o Semantics of logical connectives and quantifiers
    o Valid, satisfiable, and inconsistent sentence(s)
    o Logical consequences
    o **Be able to translate between English sentences and FOL sentences**
    o Soundness and completeness of proof theory in FOL