CS 171 QUIZ 1 prepare

properties of task environments

Fully observable vs. partially observable

!!**Fully observable**: Sensors give complete state of environment at each time point
partially observable: an environment might b e partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data
unobservable: if the agent has no sensors at all then the environment is unobseravable

Agents:

Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, based on the evidence provided by the percept sequence and whatever built-in knowledge the agent has.
!!**Performance measure:** An objective criterion for success of an agent's behavior ("cost", "reward", "utility")
omniscience: all-knowing with infinite knowledge
autonomous: An agent is autonomous if its behavior is dtermined by its own percepts & experience (with ability to learn and adapt) without depending solely on build-in knowledge

single agent vs. multiagent

Agent: Perceives environment by sensors, acts by actuators
!!**Multiagent**: More than one agent in the task environment

deterministic vs. stochastic

!!**deterministic:**Nex state is exactly determined by current state and agent action
!!**Stochastic**: Next state not exactly determined by current state and agent action
!!**Uncertain**: Not fully observable or deterministic
nondeterministic: actions are characterized by their possible outcomes, but no probabilities are attached to them.

episodic vs. sequential

!!**Episodic**: A series of atomic episodes, each independent of prior agent actions
!!**Sequential**: The current decision could affect all future decisions

static vs. dynamic

!!**Static**: Environment does not change while the agent is deliberating
!!**Semidynamic**: Environment does not change while the agent is deliberating, but its performance measure does
!!**Dynamic:** Environment can change while the agent is deliberating

discrete vs. continuous

!!**Discrete**: Finite number of states, percepts, and actions

continuous: infinite number of states, percepts, and actions

known vs. unknown

!!**Known**: The outcomes (or probabilities) for all actions are given
unknown: the agent have to learn how it works in order to make good decisions

Task environment (PEAS)

- performance (measure)
- environment
- actuators
- sensors

search properties

Strategies are evaluated along the following dimensions:

- completeness: does it always find a solution if one exists?
- time complexity: number of nodes generated
- space complexity: maximum number of nodes in memory
- optimality: does it always find a least-cost solution?

Time and space complexity are measured in terms of

- b: maximum branching factor of the search tree
- d: depth of the least-cost solution
- m: maximum depth of the state space (may be ∞)
- (for UCS: C*: true cost to op;mal goal; ε > 0: minimum step cost)

search alg	chracteristics	Complete?	Time complexity	Space complexity	Optimal?
Depth-First	Frontier = Last In First Out (LIFO) queue; Goal-Test when inserted.	No: fails in loops/infinite-depth spaces	$O(b^m)$ with m =maximum depth of space	$O(bm)$, i.e., linear space!	No: It may find a non-op;mal goal first
Breadth-First	FIFO, goal test after node is popped off	yes, it always reaches a goal	$O(b^d)$	$O(b^d)$(keeps every node in memory, either in frontier or on a path to frontier)	Yes. It is only optimal if path cost is a non-decreasing function of depth, i.e. $f(d) \geq f(d-1)$
Uniform-Cost	goal test after node is popped off. FIFO, Frontier = queue ordered by path cost. Equivalent to breadth-first if all step costs all	Yes, it b is finite and step cost $\geq \varepsilon > 0$ (otherwise it can get stuck in infinite	$O(b^{\lfloor 1+C^*/\varepsilon \rfloor}) \approx O(b^{d+1})$	$O(b^{\lfloor 1+C^*/\varepsilon \rfloor}) \approx O(b^{d+1})$	Yes, for any step cost $\geq \varepsilon > 0$.

	equal.	loops)			
Depth-Limited	goaltest when inserted. Only search until depth L	No	$O(b^l)$	O(bl)	No
iterative deepening	Goal test when inserted.Increase depth iteratively. - Inherits the memory advantage of DFS; - Has the completeness property of BFS	Yes	$O(b^d)$	$O(bd)$	Yes, if cost is a non-decreasing function only of depth.
bidirectional (if applicable)	simultaneously search forward from S and backwards from G \n – stop when both “meet in the middle” \n – need to keep track of the intersection of 2 open sets of nodes	Yes	$O(2b^{(d/2)}) = O(b^{(d/2)})$	$O(2b^{(d/2)}) = O(b^{(d/2)})$	Yes
greedy best-first search	Same for differnt goal test strategy. $h(n)$	no(tree)	$O(b^m)$	$O(b^m)$	no
A*search	goal test after node is popped off. $f(n) = g(n) + h(n)$	yes	$O(b^m)$	$O(b^m)$	Yes. With: Tree-Search, admissible heuristic; Graph-Search,consistent heuristic

heuristic searach

admissible:

A heuristic h(n) is admissible if for every node n, $h(n) \leq h^*(n)$.

h*(n): the true cost to reach the goal state from n

consistent:

A heuristic is consistent (or monotone) if for every node n, every successor n’ of n generated by any action a,
$h(n) \leq c(n, a, n') + h(n')$