Timed Practice Exam 1
(Python, Agents, Search, Games, CSPs, Logic)

- Time limit: 75 minutes   Total: 100 points
- Show work where requested. Circle choices for MCQs.
- A single 1-page notesheet is allowed. No electronic devices.

---

1. Python Review

[14 points]

(1) What is the output? [**3** pts]

```
def f(xs, k=2):
    ys = xs[:]          # copy
    ys.append(k)
    return sum(y for y in ys if y % k == 0)

print(f([1,2,3], k=3))
```

(2) What does this print? [**3** pts]

```
def g(a, L=None):
    if L is None: L = []
    L.append(a)
    return L

x = g(1); y = g(2); z = g(3, [])
print(x, y, z)
```

(3) Circle *all* true statements. [**4** pts]

(a) `tuple` is immutable; `list` is mutable.

(b) `dict` keys must be hashable.

(c) `set([1,2,2,3])` has length 4.

(d) `'a' in {'a': 1, 'b': 2}` is `True`.

(4) Write a one-liner to produce a list of squares for even x in `range(10)`. [**4** pts]

2. Rational Agents

[12 points]

(5) Define a rational agent in one sentence. [**3** pts]

(6) Circle the correct PEAS for a web search agent. [**3** pts]

(a) Performance: precision/recall; Environment: web; Actuators: ranked results; Sensors: user queries.

(b) Performance: page rank only; Environment: robot grid; Actuators: wheels; Sensors: cameras.

(c) Performance: F1 of results; Environment: web; Actuators: result ordering; Sensors: query logs.

(7) Circle all that can make an environment more difficult. [**3** pts]

(a) Partial observability

(b) Stochastic dynamics

(c) Single agent only

(d) Continuous state/action

(8) Briefly: Model-based vs. goal-based vs. utility-based agents. [**3** pts]

3. Uninformed Search

[14 points]

(9) Circle *both* complete & optimal under unit step costs. [**4** pts]

(a) BFS

(b) DFS

(c) Uniform-Cost Search

(d) Iterative Deepening (on depth, unit cost)

(10) Branching factor $b = 3$, shallowest goal at depth $d = 4$. Give time/space complexities of BFS. [**4** pts]

(11) Define *completeness* and *optimality* for search. [**3** pts]

(12) Why can DFS be incomplete? Give a typical failure mode. [**3** pts]

4. Informed Search (Heuristics & A*)

[18 points]

(13) Define **admissible** and **consistent** heuristics. [**4** pts]

(14) Suppose h is admissible but *not* consistent. Is A* still optimal? Explain briefly. [**4** pts]

(15) Weighted A*: $f(n) = g(n) + w\,h(n)$ with $w > 1$. Circle true. [**4** pts]

(a) Always optimal

(b) Completeness depends on details but often holds for positive costs

(c) Expands fewer nodes than A* (typically)

(d) Trades solution quality for speed

(16) Manhattan distance for 8-puzzle is admissible. Give 1 sentence why. [**3** pts]
(17) IDA* vs A*: one advantage and one disadvantage. [**3** pts]

5. Adversarial Search (Games)

[14 points]
(18) Define minimax value of a game state. [**3** pts]
(19) Alpha–beta pruning: circle all true. [**4** pts]

(a) Never changes the minimax value.

(b) Order of move evaluation can affect amount of pruning.

(c) Guarantees exploring only $O(b^{d/2})$ nodes in worst case.

(d) Can prune even if the exact values of some nodes are unknown.

(20) You have a depth limit and an evaluation function. What is one standard way to mitigate horizon effects? [**3** pts]
(21) In a zero-sum deterministic game with perfect information, why is expectimax *not* appropriate? [**4** pts]

6. Constraint Satisfaction Problems (CSPs)

[16 points]
(22) Define a CSP (variables, domains, constraints). [**3** pts]
(23) Circle all that are standard CSP speedups. [**4** pts]

(a) MRV (minimum remaining values)

(b) Most constraining variable (degree heuristic)

(c) Least constraining value

(d) Random restarts only

(24) Briefly: what does *forward checking* do during backtracking search? [**3** pts]
(25) Arc consistency (AC-3) in one sentence, and its worst-case complexity in terms of n variables and domain size d. [**3** pts]
(26) Local search with min-conflicts: state representation and move rule. [**3** pts]

7. Logical Agents (Propositional Logic)

[12 points]

(27) Entailment vs. inference: define $\models$ and $\vdash$. [**3** pts]
(28) Soundness and completeness in one line each. [**3** pts]
(29) Put the formula into CNF: $(P \Rightarrow Q) \Leftrightarrow R$. [**3** pts]
(30) **Resolution** (one step): from $(A \vee B)$ and $(\neg B \vee C)$ derive ____________. [**3** pts]

8. Mixed Short Problems

[10 points]

(31) Python: What does this print? [**3** pts]

```
def h(xs):
    return {x for x in xs if xs.count(x) == 1}
print(sorted(h([3,1,2,3,2,4])))
```

(32) Heuristics: Give one reason an admissible heuristic can still be weak in practice. [**3** pts]
(33) Games: Name one technique (besides alpha–beta) that improves practical play strength with a fixed time budget. [**4** pts]

End of Exam

Answer Key (Concise)

1. 6

1 [2] [3]

2. (a), (b), (d)
3. `[x*x for x in range(10) if x % 2 == 0]`
4. Maximizes expected performance given percepts/knowledge.
5. (a) *(c acceptable)*
6. (a), (b), (d)
7. Model-based: internal state; Goal-based: achieve goals; Utility-based: maximize expected utility.
8. (a), (d)
9. Time $O(3^4) = O(81)$; Space $O(81)$
10. Completeness: finds a solution if one exists. Optimality: finds least-cost solution.
11. May follow infinite path / get stuck in cycles; fails to backtrack to shallow solutions.
12. Admissible: $h \leq h^*$. Consistent: $h(n) \leq c(n, a, n') + h(n')$.
13. Not guaranteed under closed-list graph search; need re-open for optimality.
14. (b), (c), (d)
15. Tiles must move at least their Manhattan distance (lower bound).
16. Adv: low memory; Disadv: more node expansions.
17. Utility under optimal play (MAX vs MIN).
18. (a), (b), (d)
19. Quiescence search.
20. Adversarial setting; use minimax (not expectation).
21. Variables, domains, constraints; find consistent complete assignment.
22. (a), (b), (c)
23. Prune neighbor domains after assignment; fail early on empty domain.
24. Each value must be supported in neighbors; $O(n^2 d^3)$.
25. State: complete assignment; Move: pick conflicted var, assign min-violations value.
26. $\models$: semantic entailment; $\vdash$: syntactic derivation.
27. Sound: $\vdash \Rightarrow \models$; Complete: $\models \Rightarrow \vdash$.
28. $(P \lor R) \land (\neg Q \lor R) \land (\neg R \lor \neg P \lor Q)$
29. $(A \lor C)$
30. `[1, 4]`
31. Too low/informative $\Rightarrow$ behaves like UCS (many expansions).
32. Move ordering (e.g., history/killer moves) *(any valid example OK)*.