COL 333/671 Autumn 15, Minor 1

Welcome to minor 1. The exam is for 1 hour. Questions numbered 1-20 are two points each unless otherwise stated. Questions may have multiple correct choices.

Please use only pens while answering questions. Do not use a pencil.

Question Number	Maximum Marks	Marks Obtained
1-20	40	30
21	10	3
22	10	2
23	10	5
24	15	4.5
25	15	5

49.5

1. Early methods in AI were heavily based upon

A

(A) logic
(B) probability

2. Hill climbing with random restarts is guaranteed to find the global optimum if it runs long enough on a finite state space.

A

(A) True
(B) False

3. Identify all differences between simulated annealing (SA) and genetic algorithms (GA)

B C A

(A) GA maintains multiple candidate solutions
(B) GA can usually adapt well if the optimization function is suddenly changed in the middle of the algorithm run, but SA usually does not.
(C) SA has no parameters to set whereas GA has several.
(D) GA will always converge to an optimal solution faster than SA on any given problem.

4. Stochastic local beam search is equivalent to multiple runs of greedy hill climbing with random restarts.

B

(A) True
(B) False

5. The number of mutexes in each proposition layer in planning graph monotonically decrease.

T

(A) True
(B) False

6. Planning graph heuristic also computes an admissible heuristic for a temporal planning problem where the objective is to minimize the total time to reach a goal, as long as no action takes less than 1 unit time.

T

(A) True
(B) False

7. The famous classical planner FF uses ______ as its base search algorithm.

B

(A) Heuristic search
(B) Enforced hill climbing
(C) Iterative deepening search
(D) Greedy hill climbing with random restarts

8. The presence of non-mutex goal literals in planning graph implies that the goal is reachable

(A) True
(B) False

F

9. If I knew that the mini-max value of the root is between $x - \epsilon$ and $x + \epsilon$, how will I modify the mini-max algorithm to use this information

(A) Set $\alpha = x - \epsilon$ and $\beta = x + \epsilon$ at the root
(B) Set $\alpha = x + \epsilon$ and $\beta = x - \epsilon$ at the root
(C) This information cannot be used directly by mini-max.

A

10. Suppose I replace every min node in mini-max tree with a chance node with arbitrary probability distribution and evaluate it using expecti-max instead of mini-max. Let the root node be a max node. Then the backed-up value of original minimax root node will be no less than the value of chance-modified tree's root node.

(A) True
(B) False

T

11. Which of the following are used to mitigate the horizon effect in adversarial search

(A) Quiescence search
(B) Secondary search
(C) Transposition tables

A B C

12. Which of the following games have been solved till date?

(A) Chess
(B) Checkers
(C) Go
(D) Tic-Tac-Toe

A D B

13. If h_1 and h_2 are two admissible heuristics which of the following are guaranteed to be admissible

(A) h_1+h_2
(B) $\min(h_1,h_2)$
(C) $\max(h_1,h_2)$
(D) $\alpha.h_1 +(1- \alpha)h_2$ for $\alpha \in [0,1]$

B C D

14. Setting w=0 in Weighted A* is equivalent to

(A) A*
(B) Greedy best first search
(C) None of the above

C

15. A* search will always expand fewer (or as many) search nodes than uniform cost search.

(A) True
(B) False

B

16. If a search graph has negative edge costs, which of the following algorithms return an optimal solution

(A) Uniform cost search
(B) TREE-SEARCH A* with admissible heuristic
(C) Both of these
(D) None of these

D

17. Breadth first search is complete even in the presence of zero edge costs.

(A) True
(B) False

A

18. GRAPH-SEARCH can give exponential speedup over TREE-SEARCH

(A) True
(B) False

A

19. IDA* implementation does not need a priority queue but A* does.

(A) True
(B) False

A

20. Iterative deepening search (IDS) with depth limits of 1, 2, 4, 8, 16,... will be much more efficient than standard IDS in terms of number of search nodes explored to find a goal

(A) True
(B) False

A

21. [4 pts] Give one advantage and one disadvantage of equating artificial intelligence with beating humans in Turing Test.

Disadvantage :

- No clear benefit of Turing Test and has also lead to people developing tricks rather than actually doing something significant.

1.5

0

[4 pts] You are writing a game-player for a deterministic adversarial game and using minimax strategy to determine its moves. Give two reasons why you would program a deviation from minimax in some situations (which situations?). Comment: alpha-beta pruning is not a deviation from the minimax strategy, it's just an efficient implementation of that strategy.

X We would have to do a cutoff search in some situations if the depth of the game tree is very large. We can set the cutoff for search to be a queisent state and backup the values by using a evaluation function at the cutoff leafs.

[2 pts] Give one disadvantage of enforced hill climbing over greedy hill climbing.

1.5 Enforced hill climbing at times can get very slow as traversing state breadth wise is problematic ~~interms~~ of time and space.

[2 pts] Give one advantage of genetic algorithms over greedy hill climbing.

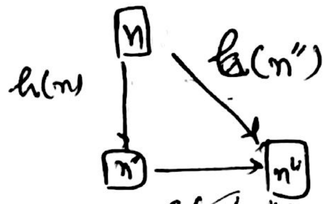
Genetic algorithms do not intend to maximize a single candidate they are inclined towards increasing "mixing" which may not give best solutions always. Moreover Genetic algorithms have

X many parameters which need to be carefully chosen for effective working of the algorithm.

adv.

22. [10 points] Prove that any consistent heuristic h is also admissible. You may assume that h(G) = 0 for every goal and that all costs are non-negative.

Consistency is triangle inequality



$h(n'') < h(n) + C(n', n'')$

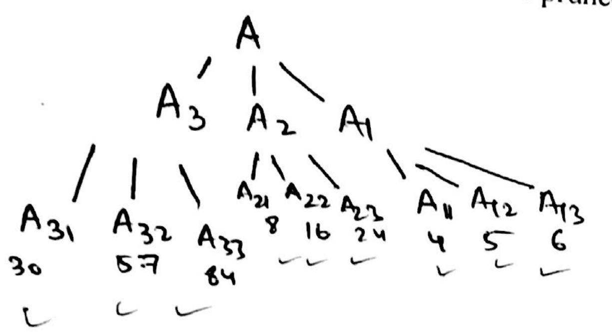
So our f(n) is monotonically increasing while admissibility implies that we never overestimate so the estimate of c(n', n'') say h' is less

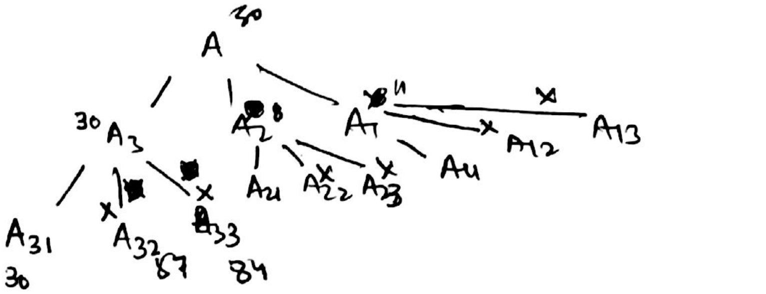
$h' < C(n', n'')$

2

23. [10 points] Mini-max search with alpha-beta pruning.

(a) In the mini-max search tree the root node A is a MAX node. It has three children A1, A2, A3, who have three children each: A11, A12, A13 (parent is A1); A21, A22, A23 (parent is A2), and A31, A32, A33 (parent is A3). Let us suppose minimax tree computes the backed-up values of Aij nodes to be $3|i-2| + i^3 j$. How should the children Ai, and Aij be ordered so that alpha-beta pruning has the best pruning performance? Draw the tree assuming children are visited left to right. Which nodes will be pruned?





A32, A33, A22, A23, A12, A13 are pruned

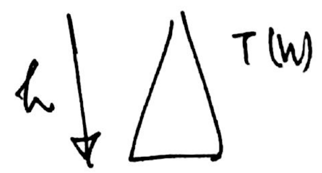
5

(b) Write a recurrence relation for mini-max with alpha-beta pruning in the best case. Assume that T(h) represents the number of nodes visited if the sub-tree has height h. Also, give the base case(s). Explain your recurrence.

$$T(h) = T(h-1) + \frac{T(h-2)}{2} + \frac{T(h-3)}{4} + \cdots + \frac{T(1)}{2^h}$$

As we get bettr backup value we can prune the last- end o the traversal.

0



may

min

max

24. [15 points] Recall the Blocks-world domain discussed in class. It has type 'block', and predicates, (on ?b1, ?b2), (on-table ?b), (clear ?b), (holding ?b), and (arm-empty). The actions are pickup(?b1, ?b2), pickup-table(?b), place(?b1, ?b2) and place-table(?b).

assumption

(a) Write the STRIPS description for the action place(?b1, ?b2) place : ?b1 on ?b2

:action place ?b1 ?b2

:precondition

- clear (?b1)
- arm-empty X -0.5 0.5
- clear (?b2) ✓ 1
-

holding (?b1)

clear (?b2)

:effect

- (on ?b1, ?b2) ✓ 1
- ~~(not clear ?b2)~~ 3
- clear(?b1)
- arm-empty ✓ 1
- not (clear ?b2) ✓ 1

1.5

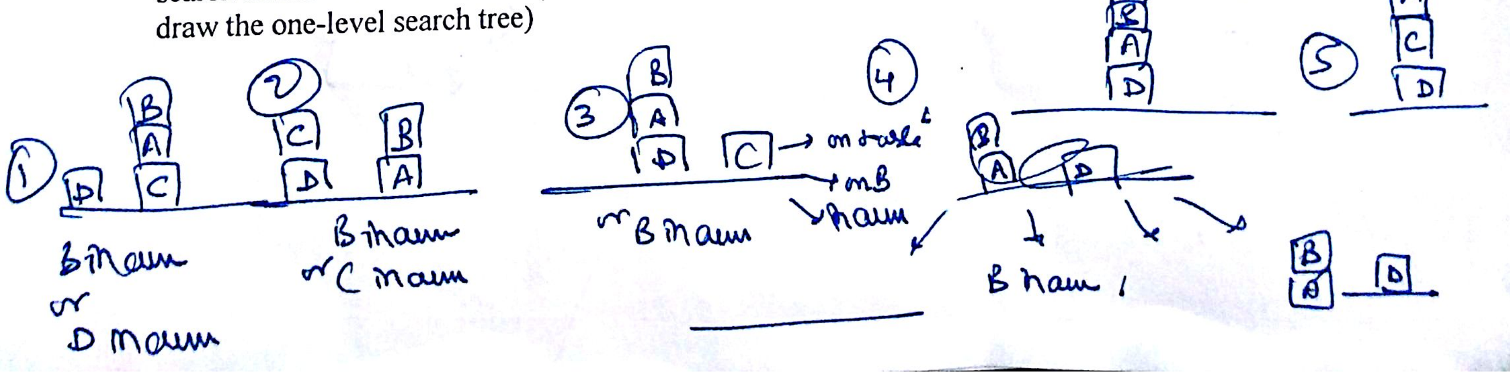
~~not clear is a required predicate~~

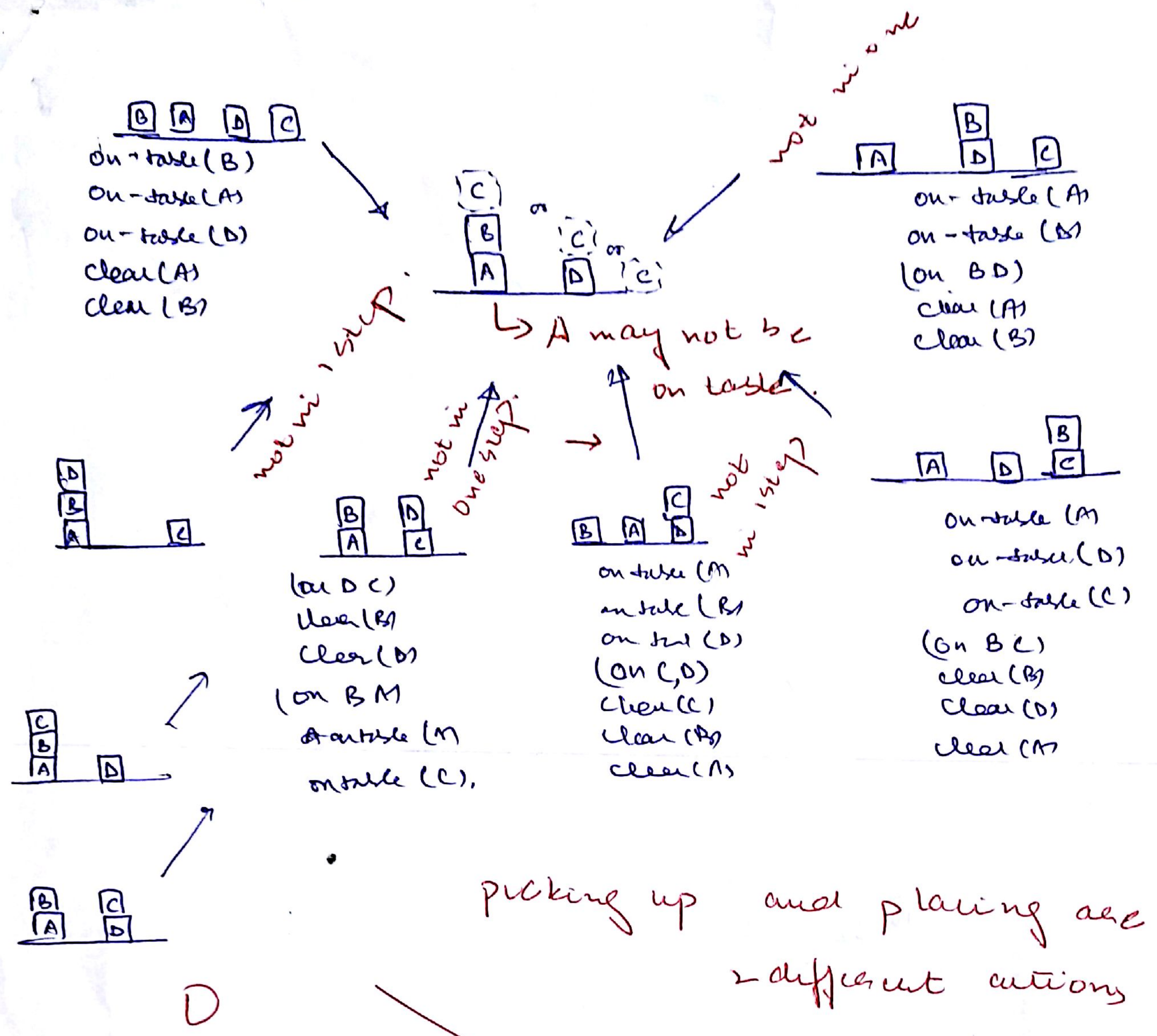
or

~~State clear (?b2) is not the effect~~

(on-table D) ∧ (on B A) ∨ (on A C) ∨ (on A ~~table~~ free A) ∨ (on-table D) ∨ (on-table C)

(b) Suppose our final goal is {(on-table D), (on B A)}. Here, A, B, C, D are constants of the type 'block'. If we were to do subgoal space search in backward direction, how many (grounded) actions would be present at the topmost level of the search tree? Which ones? What will be the search nodes at the first level of search tree through each of these actions? (You may simply draw the one-level search tree)





25. [15 points] Pacman and Ms. Pacman are lost in an 10x10 maze and would like to meet; they don't care where. In each time step, both simultaneously move in one of the following directions: {NORTH, SOUTH, EAST, WEST}. Trying to move to a wall or a blocked location forces them to stay in their original location. They do not alternate turns. You know both their locations and must devise a plan for them that positions them together, somewhere, in as few time steps as possible. Passing each other does not count as meeting; they must occupy the same square at the same time.

(a) [5 points] Formally state this problem as a single-agent state space search problem. Define state space, action space, initial state, goal test and edge costs. You need not define the successor function.

action space : move north Packman
move south Packman
move east Packman
move west Packman.

goal test : location[Pacman] = location[Ms. Pacman] & time is same for both.

edge costs : unit

2

(b) [2 points] What is the maximum size of a state space?

$^{100}C_2$ ✗

(c) [2 points] What is the maximum branching factor?

8 ✗

(d) [3 points] Give a non-trivial and informative admissible heuristic for this problem.

Straight line distance between the Packman & Ms. Packman ✗

(e) [5 points] Which of the following graph search algorithms are guaranteed to output an optimal solution to this problem.

(i) Uniform cost search

(ii) DFS

(iii) BFS

(iv) Greedy best first search with consistent heuristic

(v) A* with consistent heuristic

(vi) A* with zero heuristic

3