CSE 415 Autumn 2023 Assignment 4

Last name:______________________ First name:_________________________________

UWNetID: ______________________

Due Wednesday night November 1 via Gradescope at 11:59 PM. You may turn in either of the following types of PDFs: (1) Scans of these pages that include your answers (handwriting is OK, if it's clear), or (2) Documents you create with the answers, saved as PDFs. When you upload to GradeScope, you'll be prompted to identify where in your document your answer to each question lies.

Do the following six exercises. These are intended to take 25-35 minutes each if you know how to do them, except that Exercise 4 may take additional time in order to read the relevant portions of the Samuel paper. Each is worth 25 points. If any corrections have to be made to this assignment, these will be posted in ED.

This is an individual-work assignment. Do not collaborate on this assignment.

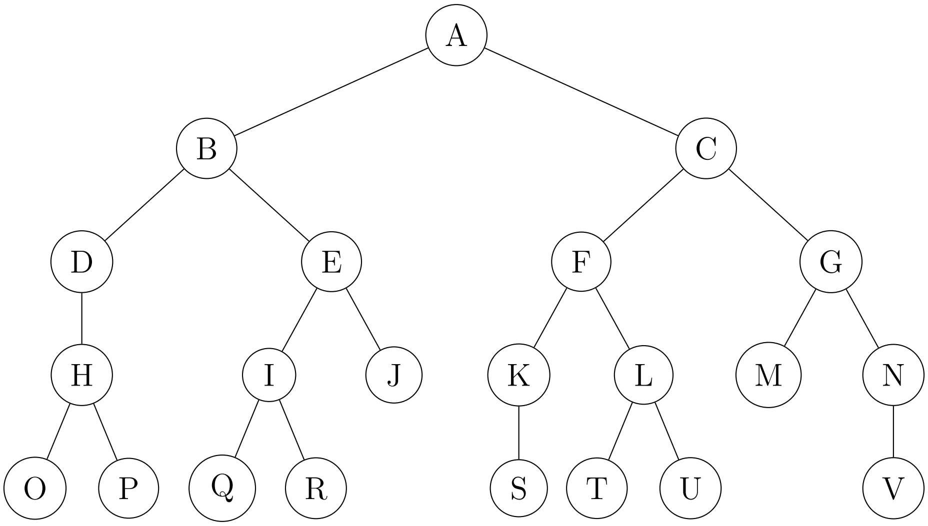
Prepare your answers in a neat, easy-to-read PDF. Our grading rubric will be set up such that when a question is not easily readable or not correctly tagged or with pages repeated or out of order, then points will be be deducted. However, if all answers are clearly presented, in proper order, and tagged correctly when submitted to Gradescope, we will award a 5-point bonus.

If you choose to typeset your answers in Latex using the template file for this document, please put your answers in blue while leaving the original text black.

Version 23-10-25a. If corrections to this document are required, this will be announced in ED.

1 Basic Search

Use the following tree to answer the questions below comparing Breadth First Search, Depth First Search, and Iterative-Deepening Depth First Search. Assume that children are visited from left to right.



(a) (3 points) Write out the order that the nodes are expanded in using Breadth First Search, starting from A and searching to K.

(b) (3 points) Write out the order that the nodes are expanded in using Depth First Search, starting from A and searching to K.

(c) (3 points) Write out the order that the nodes are expanded in using Iterative-Deepening Depth First Search, starting from A and searching to K. If a node is repeated, make sure to include it each time it is expanded.

(d) (4 points) Which of the three search algorithms (BFS, DFS, IDDFS) has the smallest maximum size of the open list while searching from A to K? What is the maximum size of the open list for that algorithm?

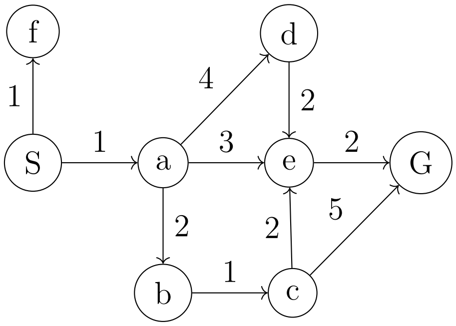
(e) (2 points) True or False: BFS, DFS, and IDDFS will each return the same path starting at A going to K.

(f) (2 points) True or False: Given the same start and goal state, BFS, DFS, and IDDFS will always return the same path for any search graph.

(g) (4 points) Under what conditions might BFS be a better choice of search algorithms when compared to DFS? (Give one possible reason, 1 sentence)

(h) (4 points) Under what conditions might IDDFS be a better choice of search algorithms when compared to BFS? (Give one possible reason, 1 sentence)

2 Heuristic Search



For the following questions, consider three heuristics h_1, h_2, h_3. The table below indicates the estimated cost to goal, G, for each of the heuristics for each node in the search graph.

state (s)	S	f	a	b	c	d	e	G
heuristic $h_1(s)$	6	12	5	2	3	4	2	0
heuristic $h_2(s)$	6	5	6	4	4	4	1	0
heuristic $h_3(s)$	5	8	5	4	3	3	2	0

(a) (2 point) What does it mean for a heuristic to be "admissible"?

(b) (3 points) Which heuristics among $\{h_1, h_2, h_3\}$ shown above are admissible? Identify any violation(s) found in making this determination.

(c) (2 point) What does it mean for a heuristic to be "consistent"?

(d) (3 points) Among the heuristics you identified as admissible in part (a), which are also consistent? Identify any violations found in making this determination.

(e) (4 points) Show the node expansion order going from S to G using $A*$ with h_1. Also provide the final path obtained.

(f) (4 points) Show the node expansion order using $A*$ for h_3.

(g) (2 points) Of the heuristics you used in (e) and (f) above, which would you consider the better heuristic? Explain why, using examples from your attempts to use each heuristic to support your explanation.

(h) (2 points) What does it mean for one heuristic to dominate another heuristic?

(i) (1 points) Consider h_1 – what change(s) would you make to this heuristic to make it the dominant heuristic of the three options available? Describe both the change you'd make and how the heuristic now satisfies the conditions for dominance.

(j) (2 points) Explain why, when using the $A*$ search algorithm, it is important to continue the expansion process until the goal state is removed from the OPEN list and becomes the current state, rather than terminating as soon as the GOAL sate is found.

3 Designing Heuristic Functions

As the midterm season is coming up, you as the most senior student want to create an informative welcome video for the incoming first-year to explain about how to navigate the school. Based on your many years experience as a student, you have realized that new students do not like to listen to a long talk given by a single person. However, as a student studying computer science, you are too lazy to create a video yourself. So, you have come up with an idea that you will instead be just combining video footage from previous years' recordings.

More formally, this exercise looks into the use of state-space search to design an *optimal* video mix. Suppose there is a dataset $V = \{v_1, v_2, \ldots, v_n\}$ of video segments, together with information about segment lengths in seconds and the topics covered. The *optimal* video is defined to be a set of of video segments that cover the set of specific topics while still using the least total of segment lengths. Note that the order of video segments does not affect the optimal conditions.

Segment Name	**Segment Length (seconds)**	**Topics covered**
`2023au`	10	`[welcome]`
`2023su`	30	`[skiing, views]`
`2023sp`	50	`[welcome, AI, ChatGPT]`
`2023wi`	40	`[salmon, dragons, skiing]`
`2022au`	50	`[skiing, ChatGPT]`

Table 1: Dataset of video segments

For simplicity, we formulate this problem as a graph search by representing states (and a node for each state) as $s_i = (r_i, m_i)$ where m_i is the list of remaining topics to be covered and m_i represents the mix so far: a list of video segments that have been added. Note that none of these video segments in m_i has any topics in r_i, since those topics are the ones that still need to be covered by some additional video segment.

The set of operators $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$ has one operator for each video segment; the operator ϕ_j means to add the j^{th} video segment to the mix. The precondition for ϕ_j is that the j^{th} video segment covers (among other possible topics) the first element of r_i. The state-transformation for ϕ_j is to remove any topics from r_i that are covered by the j^{th} video segment and to add this video segment to the mix m_i.

Thus, for any state $s_i = (r_i, m_i)$, its successors are found by considering all video segments not in m_i that contain the first topic from r_i. For instance, the successors of state $(\texttt{[welcome, dragons]}, \texttt{[]})$ are $(\texttt{[dragons]}, \texttt{[2023au]})$ and $(\texttt{[dragons]}, \texttt{[2023sp]})$. The weight of the edge from s_i to the successor produced by ϕ_j is defined to be the segment length of the j^{th} video segment.

(a) (10 pts) For this assignment, assume the initial state is
$s_{\texttt{initial}} = (\texttt{[welcome, skiing, ChatGPT]}, \texttt{[]})$.
Draw the portion of the search space reachable from this state.

(b) (5 pts) Give a non-trivial admissible heuristic function $h(s)$. (This could then be used with an A* search for this problem, and in theory, much larger examples of this sort of problem.) Note that $h(s) = 0$ is the trivial heuristic function.

(c) (5 pts) Prove that your heuristic function is admissible.

(d) (5 pts) Consider the following heuristic function. Is the heuristic function admissible? If yes, show the proof. Otherwise, please provide one or more counterexamples.

$$h(s) = h\left((r, m)\right) = \sum_{t \in r} q(t)$$

where $q(t)$ is the length of the smallest video segment that covers topic t.

4 Adversarial Search: Static Evaluation and Minimax

For this problem, we will be referring to the paper "Some Studies in Machine Learning Using the Game of Checkers"[1] written by Arthur L. Samuel at IBM in 1959. It is one of the pioneering works not only for adversarial search, but also for modern machine learning algorithms.

Please read the 'Introduction' and 'The basic checker-playing program' sections. Then read the following (sub)sections to answer the questions below: 'Ply limitations,' 'Rote Learning and its variants,' 'Learning procedure involving generalizations' (first two subsections). The number of the relevant page for each question is included in brackets after the question.

4.1 Warm up

(a) T/F The program evaluates a board position at a minimum look-ahead ply if the last move was a jump (p538).

(b) Why is it not advisable to take the initial move which leads to the highest scoring board position (p538)?

(c) T/F The most important part of the scoring polynomial used during play is the 'inability for one side or the other to move' (p536).

4.2 Learning

(d) In machine learning algorithms, it is common to use "regularization", an approach that leads your learner to (i) ignore some information, in order to (ii) prioritize more generalizable concepts from your training data. Which two terms in the paper allude to (i) and (ii)? Explain why they are beneficial to the learning program (p540).

(e) When attempting to generalize a game learner, a subset of coefficients from the scoring polynomial are selected for the static evaluation. However, this can sometimes lead to severely under-performing agents as can be seen by, for example, losing three turns in a row. A suggestion for such a case is to set which polynomial term to zero and why (p542)?

---

[1] (link to the paper)

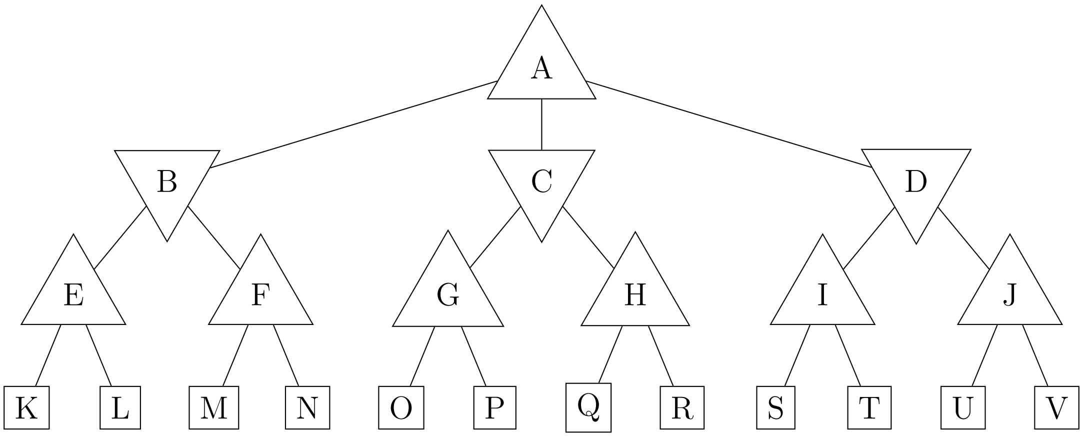
4.3 Efficiency

(f) What is one way to increase the number of backed-up plies used in selecting the next move, rather than what is typically possible without incurring much extra computation time cost (p540)?

(g) Why are the records of board positions arranged such that those likely to be seen at the beginning of the game appear earlier on tape (p540)?

5 Adversarial Search and Alpha-Beta Pruning

Use this search tree along with the table of state evaluation values to answer the questions below. By default, children are processed from left to right.



state	K	L	M	N	O	P	Q	R	S	T	U	V
evaluation	3	7	1	-5	4	8	-1	-3	-4	-2	6	2

(a) (10 points) Use minimax to perform adversarial search with alpha-beta pruning on the tree above. Fill in the values in the table below as you go. For α and β values, write the values that are passed from the state's parent to that state. For example, the value of state K will not be shown in the $\alpha - \beta$ values of state E, but would be reflected in the $\alpha - \beta$ values of state L. If a state does not have to be evaluated, do not write any values for it in the table.

state	value	α	β
A		N/A	N/A
B			
C			
D			
E			
F			
G			
H			
I			
J			
K	3		
L	7		
M	1		
N	-5		
O	4		
P	8		
Q	-1		
R	-3		
S	-4		
T	-2		
U	6		
V	2		

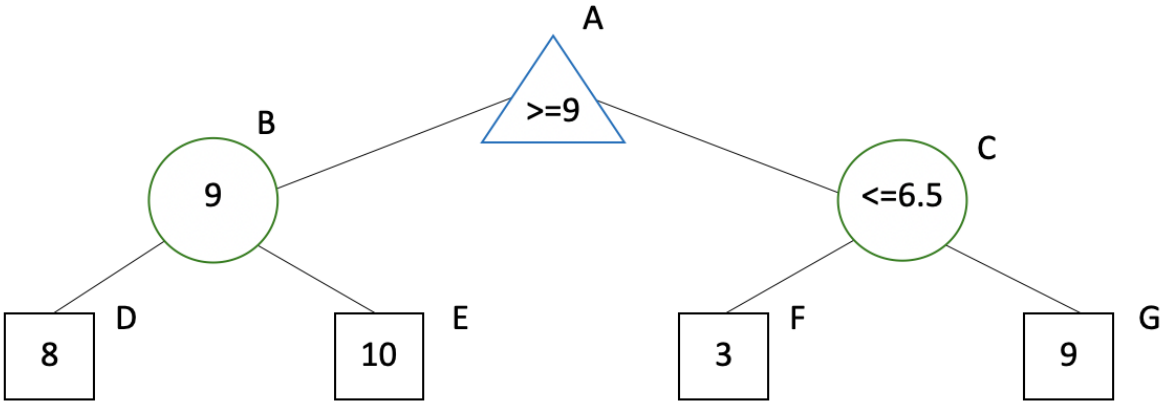
(b) (15 points) Consider optimizing alpha-beta search on this tree by re-ordering the nodes of the tree. By changing the order of the children of some nodes, the algorithm may need to evaluate more or less states due to alpha-beta pruning. What nodes can be re-ordered within the tree to minimize the number of states that need to be evaluated? How many fewer states need to be evaluated when compared to the original ordering?

When re-ordering, make sure that all parent-child relationships in the tree are unchanged, and only the order of a node's children are changed.

Hint: our staff solution makes just two changes to the ordering.

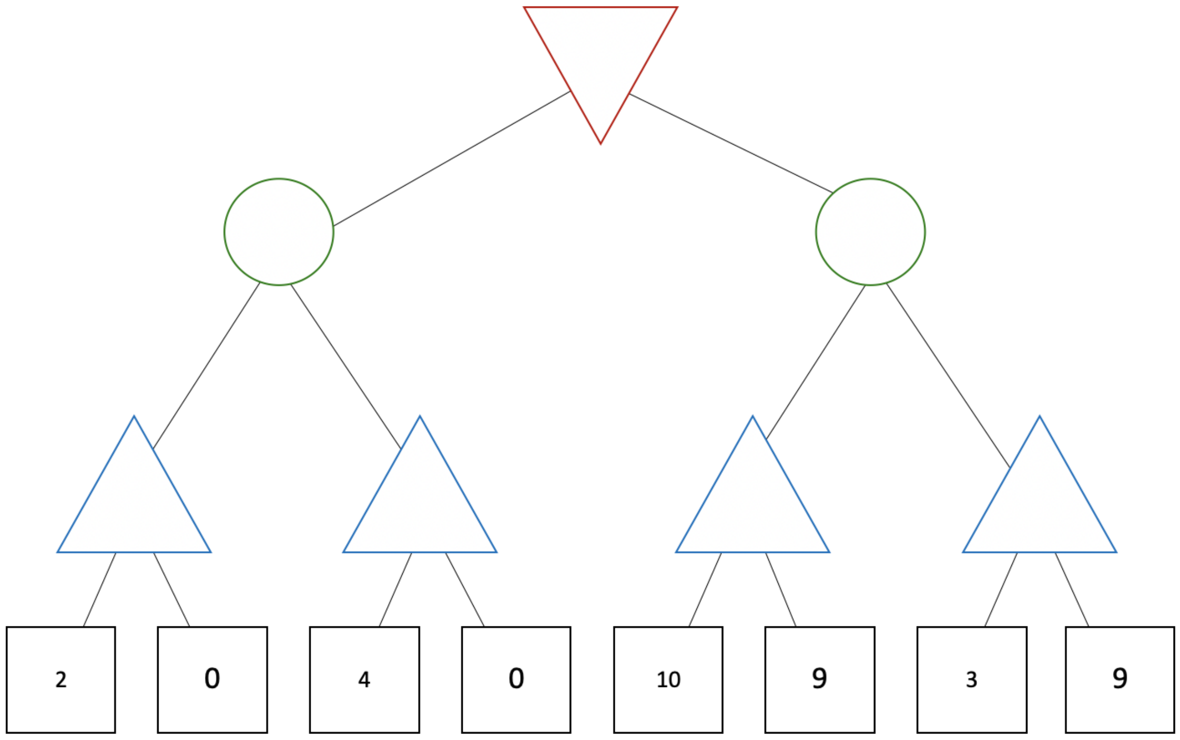
6 Pruning with Chance Nodes

Although alpha-beta pruning cannot be applied directly to searching trees that contain chance nodes, reasoning like that inherent in alpha-beta search can sometimes be applied when static values are constrained to lie within given ranges of values. For example, if all leaf-node values must be values $f(s)$ such that $0 \leq f(s) \leq 10$, then node G can be pruned in the tree below, because Max can get 9 by moving left at A to B, and if Max goes to C and finds $f(F) = 3$, Max can reason that getting $f(G)$ is useless because if $f(G) = 10$ which is larges allowed, then the value at C is $(3+10)/2 = 6.5$, which is inferior to 9.



(10 points) In the following game tree, determine where pruning can be performed using the same range assumption as above. Show where there are cutoffs

(15 points) Explain your reasoning for each cutoff.


2
0
4
0
10
9
3
9