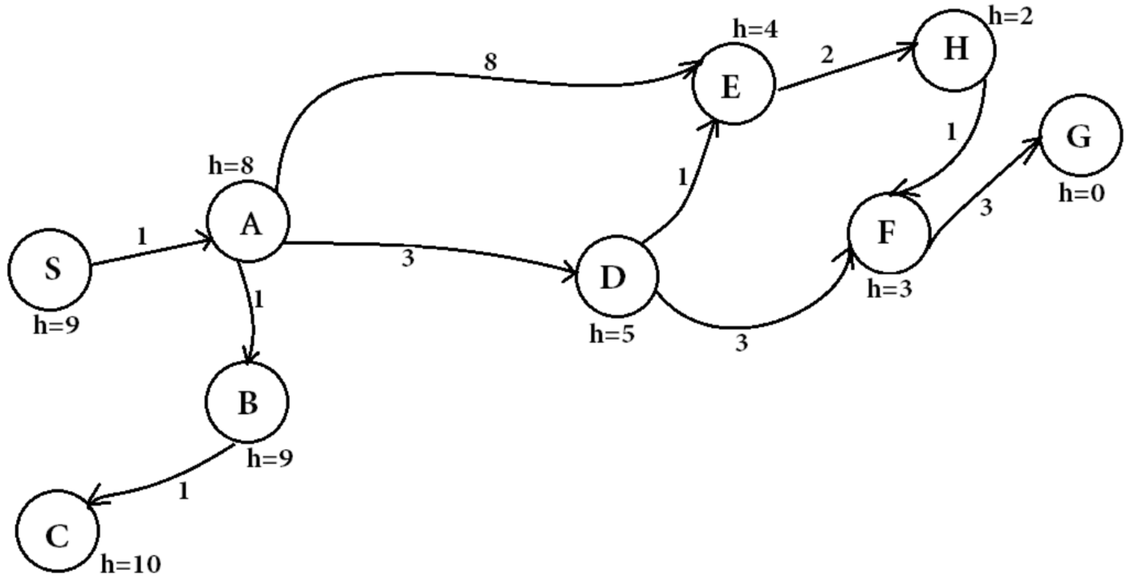
Q1 [30 pt] Search

Given the graph below, suppose you want to go from start state "S" to goal state "G", write down the order in which the states are expanded and the path found by the following search algorithms. Ties (e.g., which child to first explore in depth-first search) should be resolved alphabetically (i.e. prefer A before Z). Remember to include the start and goal states in your answer. Assume that algorithms execute the goal-test when nodes are expanded, not when their parent is expanded to generate them as successors. Assume a graph search implementation – do not expand any node more than once.



(a) [6 pt] Bread-first Search

[2 pt] Expanded order: S A B D E C F H G

[2 pt] Solution: S A D F G

[2 pt] Path cost: 10

(b) [5 pt] Depth-first Search

[2 pt] Expanded order: S A B C D F G (S A B C D E H F G)

[2 pt] Solution: S A D F G (S A D E H F G)

[2 pt] Path cost: 10 (11)

(c) [6 pt] Uniform Cost Search

[2 pt] Expanded order: S A B C D E F H G

[2 pt] Solution: S A D F G

[2 pt] Path cost: 10

(d) [6 pt] Greedy (Best-first) Search

[2 pt] Expanded order: S A E H F G

[2 pt] Solution: S A E H F G

[2 pt] Path cost: 15

(e) [6 pt] A* Search

[2 pt] Expanded order: S A D E H F G

[2 pt] Solution: S A D F G

[2 pt] Path cost: 10

Q2 [15 pt] Search

Given an encrypted file, you are trying to recover a password to it by using search algorithms. The password of the file is up to 10 letters long and contains only the letters A, B, C. The formulated search problem can be seen blow:

- The initial state is the empty string.
- The successor function is to append one letter (A, B, or C) to the string.
- The goal test is to verify a candidate password using the decryption software. There are six correct passwords:

AAACCC, ABBCC, BABAB, BCABACB, CBAC, and CBACB.

• Assume that all ties are broken alphabetically. For example, if there is a tie between states “A", “B", and ”C", expand “A" first, then “B", then “C".

a. [5 pts] From the six correct passwords below, select the one that will be returned by depth-first search:

- ○ AAACCC
- ○ ABBCC
- ○ BABAB
- ○ BCABACB
- ○ CBAC
- ○ CBACB

b. [5 pts] From the six correct passwords below, select the one that will be returned by breadth-first search:

- ○ AAACCC
- ○ ABBCC
- ○ BABAB
- ○ BCABACB
- ○ CBAC
- ○ CBACB

c. [5 pts] You suspect that some letters are more likely to occur in the password than others. You model this by setting cost(A) = 1, cost(B) = 2, cost(C) = 3. From the six correct passwords below, select the one that will be returned by uniform cost search using these costs:

- ○ AAACCC
- ○ ABBCC
- ○ BABAB
- ○ BCABACB
- ○ CBAC
- ○ CBACB

Q3 [20 pt] Local Search- Simulated Annealing

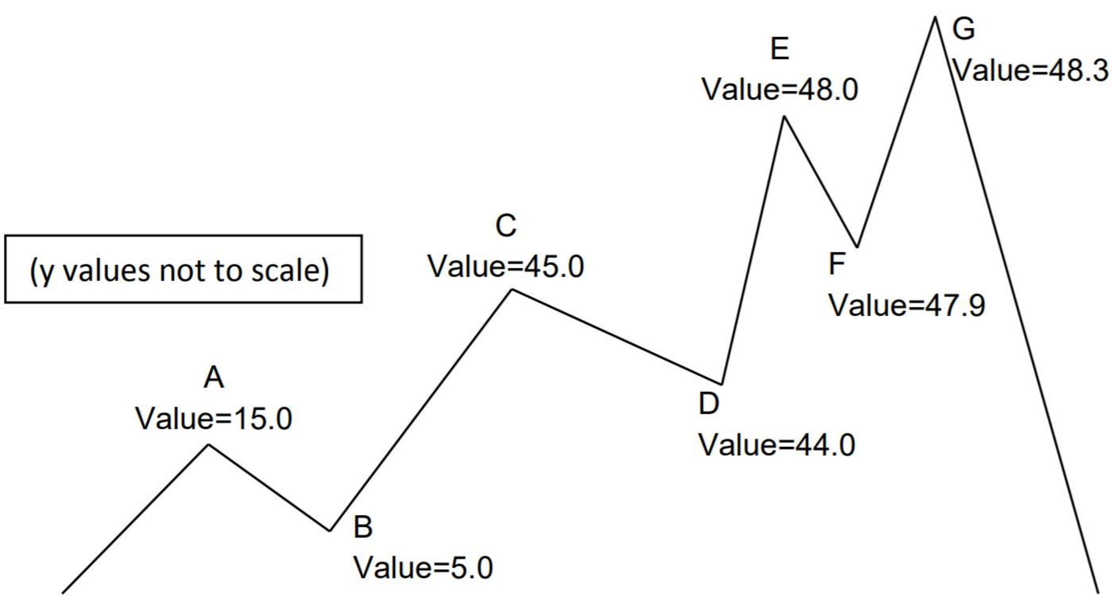
In the value landscape cartoon below, you will be asked about the probability that various moves will be accepted at different temperatures. Recall that Simulated Annealing always accepts a better move (ΔValue = Value[next] – Value[current] > 0.0); but it accepts a worse move (ΔValue < 0.0) only with probability e^(ΔValue/T), where T is the current temperature on the temperature schedule.

Please use this temperature schedule (usually, it is a decaying exponential; but it is simplified here):

Time (t)	1-100	101-200	201-300
Temperature (T)	10.0	1.0	0.1

You do not need a calculator; the values given have been chosen to follow this table:

x	0.0	-0.1	-0.4	-1.0	-4.0	-40.0
e^x	1.0	≈0.90	≈0.67	≈0.37	≈0.02	≈ $4.0e$-18



Give your answer to two significant decimal places. **The first one is done for you as an example.**

(example) You are at Point A and t=23. **The probability you will accept a move A -> B = <u>0.37</u>**

a-g: each has 2 points, h has 4 points

a. You are at Point B and t=23. **The probability you will accept a move B -> C = (1.0)**

b. You are at Point C and t=123. **The probability you will accept a move C -> B = (4.0e-18)**

c. You are at Point C and t=123. **The probability you will accept a move C -> D =(0.37)**

d. You are at Point E and t=123. **The probability you will accept a move E -> D =(0.02)**

e. You are at Point E and t=123. **The probability you will accept a move E -> F =(0.90)**

f. You are at Point G and t=123. **The probability you will accept a move G -> F =(0.67)**

g. You are at Point G and t=223. **The probability you will accept a move G -> F =(0.02)**

h. With a very, very, very long slow annealing schedule, are you more likely, eventually in the long run, to wind up at point A or at point G? (write A or G): **(G)**

Q4 [25 pt]. Q4 [20 pt]. CSPs

You are designing a menu for a special event. There are several choices, each represented as a variable:

(A)ppetizer, (B)everage, main (C)ourse, and (D)essert. The domains of the variables are as follows:

A: (v)eggies, (e)scargot

B: (w)ater, (s)oda, (m)ilk

C: (f)ish, (b)eef, (p)asta

D: (a)pple pie, (i)ce cream, (ch)eese

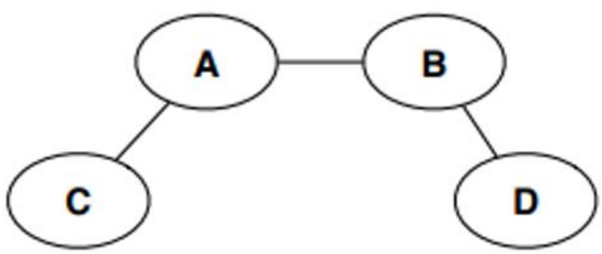
Because all of your guests get the same menu, it must obey the following dietary constraints:

(i) Vegetarian options: The appetizer must be veggies or the main course must be pasta or fish (or both).

(ii) Total budget: If you serve the escargot, you cannot afford any beverage other than water.

(iii) Calcium requirement: You must serve at least one of milk, ice cream, or cheese.

a. [5 pt] Draw the constraint graph over the variables A, B, C, and D.



b. [5 pt] Imagine we first assign A=e. Write down the updated domains of the variables after forward checking. [Each variable has a correct domain gets 1 pt, otherwise 0 pt]

Variable	Domain
A	{e}
B	{w}
C	{f, p}
D	{a , i, ch}

c. [5 pt] Again imagine we first assign A=e. Write down the updated domains of the variables after arc-consistency has been enforced. [Each variable has a correct domain gets 1 pt, otherwise 0 pt]

Variable	Domain
A	{e}
B	{w}
C	{f, p}
D	{ i, ch}

d. [5 pt] Give a solution for this CSP or state that none exists. [Each variable has a correct solution gets 1 pt, otherwise 0 pt. There are several solutions. You can check it if it is correct.]

Multiple solutions exist. One is A = e, B = w, C = f, D = i

e. [5 pt] For general CSPs, will enforcing arc consistency after an assignment always prune at least as many domain values as forward checking? Briefly explain why or why not. [Each of the two answers are ok.]

Answer: Two answers are possible:

Yes. The first step of arc consistency is equivalent to forward checking, so arc consistency removes all values that forward checking does.

No. While forward checking is a subset of arc consistency, after any assignment, arc consistency may have already eliminated values in a previous step that are eliminated in that step by forward checking. Thus, enforcing arc consistency will never leave more domain values than enforcing forward checking, but on a given step, forward checking might prune values than arc consistency by pruning values that have already been pruned by arc consistency

Q5 [10 pt]. Games.

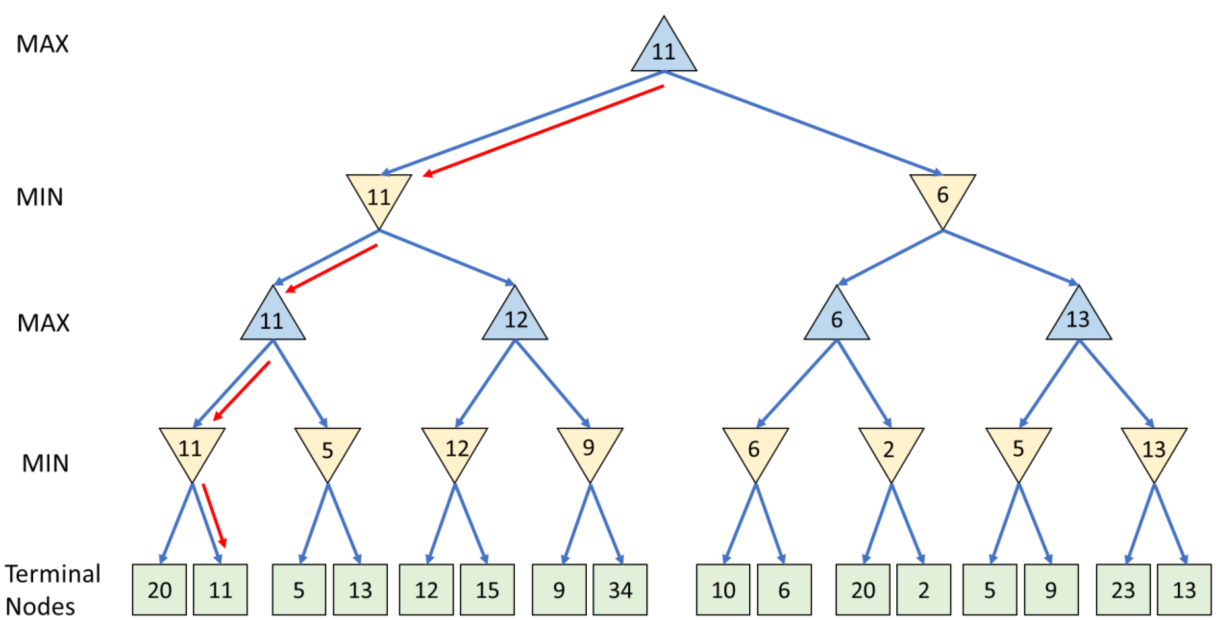
Consider the following mini-max tree, whose root is a max node. Assume that children are explored left to right. The minimax algorithm with alpha-beta pruning is used to determine the optimal moves if MAX and MIN make perfect moves.

a [5 pt] Fill in the mini-max values for each of the nodes in the tree that aren't leaf nodes resulting from a basic minimax search.

Each value in the node: 0.5 point

b [5 pt] Highlight the optimal move on the game tree.

Each move: 1 point



Additionally, if alpha-beta pruning algorithm is run on this tree, cross out all the pruned branches in the tree.