Quiz 12: Trees

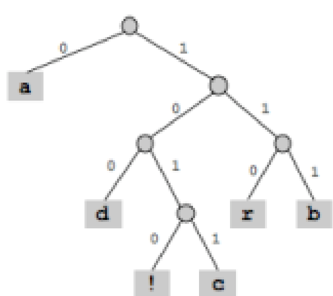
Question 1. The wheel graph W_4 has 5 vertices: 4 vertices form a cycle graph C_4 – a square; one more vertex sits in the middle and is connected with the remaining 4 vertices. (W_4 is isomorphic to an ordinary Egyptian pyramid.) Assume that all vertices in the wheel graph are named with letters and are distinguishable. Find the number of unrooted spanning trees in the W_4.
Write a positive integer in your answer.

Question 2. It is known that a full m-ary tree T has 25 leaves, but the parameter m is not known – it can take any fixed value: $m = 2, 3, 4, \ldots$. How many inner nodes can T have? Find all possible answers.
Write an increasing comma-separated list.

Question 3.
There is a rooted tree with 111 vertices and each vertex can have up to 3 children. Find the minimum and the maximum height of this tree.
Write two comma-separated integers.

Question 4.
Assume that there is a rooted tree (with anonymous/unnamed vertices and unordered children) and its vertices have the following degrees $3, 3, 2, 2, 1, 1, 1, 1$. It is also known that its root vertex has 3 children. Find the number of such rooted unordered trees.
Write a positive integer.

Question 5.



char	encoding
a	0
b	111
c	1011
d	100
r	110
!	1010

Figure 1. Encoding with a Tree.

Figure 1 shows an efficient method to send messages (like `abracadabra!`) using 6 characters. Assume that the characters appear with the following probabilities:

Symbol	a	b	d	r	c	!
Probability	1/2	1/8	1/8	1/8	1/16	1/16

Find the expected number of bits used per character (i.e. $E(X)$ – the expected value of the random variable X, which describes the number of bits used per one character from this random distribution.)
Write a real number – the number of bits rounded to the nearest thousandth.

Question 6. The string

$$\to \to \to \to \to\ a\ b\ \triangle\ \neg\ c\ \neg\ d\ c\ e\ \to\ \neg\ a\ \to\ d\ e.$$

is the prefix notation for a Boolean logic expression with one symbol replaced by a $\triangle$. What can this $\triangle$ represent?
(A) It is a propositional variable (a, b, c or similar).
(B) It is a unary Boolean operator ($\neg$ or similar).
(C) It is a binary Boolean operator ($\to$ or similar).
(D) It cannot be any of these; the expression is invalid in all these cases.
Write the answer letter (A, B, C, or D).

Question 7. Imagine that you search for all ways how to place 4 queens on a 4×4 chessboard so that they do not attack each other. (A chess queen attacks all squares on its horizontal, on its vertical and also on both diagonals.)
Imagine that you build a tree for this:
(Level 0 to Level 1) The root of the tree is an empty 4×4 chess-board; you add to it four children (all 4 ways how you can place a queen on the 1st row of the chess-board).
(Level 1 to Level 2) For any of the vertices you added in the previous step, add children on level 2 by placing another queen on the 2nd row so that it does not attack the first one.
In general, the vertex on level $L = i$ has queens on rows $1, \ldots, i$ that do not attack each other. Any vertex on level $L = 4$ will be a solution to this "4 queens problem" with all four rows containing a queen.
Write the total number of vertices in this tree (that the backtracking algorithm will visit).

Question 8. An undirected graph $G = (V, E)$ has the set of vertices V – the set of all positive divisors of the number 900 (including 1 and 900 itself). A pair of divisors (d_1, d_2) is an edge in G iff their ratio d_1/d_2 (or d_2/d_1) is a prime number.
In the root vertex $v_1 = 1$ we start the BFS (Breadth-first-search) traversal of the graph G. For every vertex we visit all its adjacent vertices in increasing order (for example edges $(1, 2)$, $(1, 3)$ and $(1, 5)$ are visited in this order. When all the children of vertex 1 are visited, we start visiting all the adjacent vertices of $v_2 = 2$, and so on.). We get the BFS traversal order like this:

$$v_1 = 1,\ v_2 = 2,\ v_3 = 3,\ v_4 = 5,\ v_5 = 4,\ \ldots$$

Find the vertices v_{13}, v_{14}, v_{15} in this BFS order.
Write three comma-separated numbers.

Question 9.

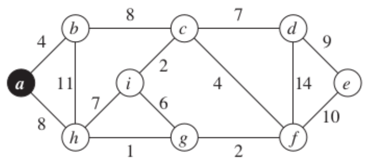


Figure 2. Weighted Graph.

Find the weight of a minumum spanning tree (MST) in the tree shown in Figure 2. In order to construct this tree, you can use Prim's algorithm (Rosen2019, p.836). Start in vertex a - this is your first tree T_1. In every step pick the minimum weight edge that is adjacent to T_i (and does not create any loop) – add it to the tree T_i, and obtain the next tree T_{i+1}. Continue adding the minimum weight edges until all vertices are connected. (In the second step you will have a choice to add (b, c) or (a, h) as both edges have the same weight 8. There may be several MSTs in a graph, but all of them will have the same total weight.)
Write the total MST weight as a number.

Question 10. The vertices in the directed graph (Figure 3) are visited in the DFS order. You start with the alphabetically smallest vertex (q), order all the vertices connected to it alphabetically, then build the DFS traversal.

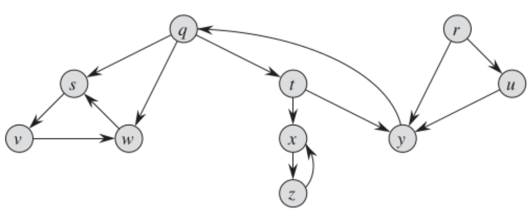


Figure 3. Graph for DFS traversal.

Write the sequence with parentheses and vertices for the graph on Figure 3 – similar to the sequence (1) – see Appendix. Each of its 10 vertices should be mentioned in your traversal order twice (the first time with an opening parenthesis, the second time with a closing parenthesis).

Appendix: DFS

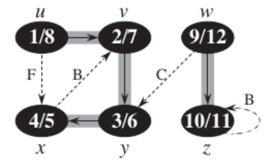


Figure 4. Sample DFS traversal.

Figure 4 shows a DFS (Depth-First-Search) traversal in a directed graph. Vertexes are visited in alphabetical order (so u is visited first; followed by its alphabetically first child v, followed by y, followed by x. After that we visit another tree (unreachable from the first one) – w followed by z.)
Tree edges that belong to the DFS traversal tree are shaded;
Back edges that point back from a vertex to its ancestor in the tree (or a loop to itself) are labeled by B;
Forward edges that jump from a vertex to its descendant in the DFS tree (other than a child) are labeled by F;
Cross edges that jump between two vertices that are not descendants/ancestors of each other are labeled by C.
The following sequence

`(u (v (y (x x) y) v) u) (w (z z) w)` (1)

denotes the DFS traversal order in the oriented graph. Every time when we enter some vertex (and its subtree), we open a parenthesis and write the vertex name; when we leave, we write the vertex name again and close the parenthesis.
This order is also written inside each vertex (for example `1/8` for vertex u means that we entered it in Step 1, and left it in Step 8). For vertex z this pair is `10/11` (we entered this leaf of the DFS tree and immediately left it).

Answers

Question 1. Answer: `45`

There are only three kinds of trees with 5 vertices, if isomorphic trees count as the same. See *isomers of pentane* – https://bit.ly/2W4BWiw (either the vertex degrees are $1, 2, 2, 2, 1$, or $1, 2, 3, 1, 1$, or $4, 1, 1, 1, 1$). Let us denote the vertices by A, B, C, D, E (E being the vertex at the top of the pyramid). The only non-existing edges are (A, C) and (B, D) (otherwise it is almost like K_5). In Figures 5, 6, 7 we show different subcases depending

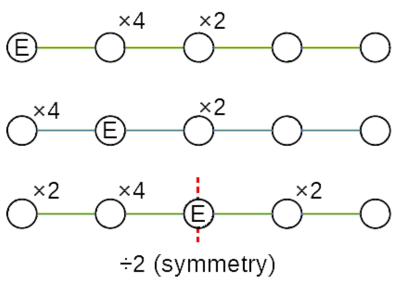


Figure 5. 24 trees with all degrees ≤ 2.

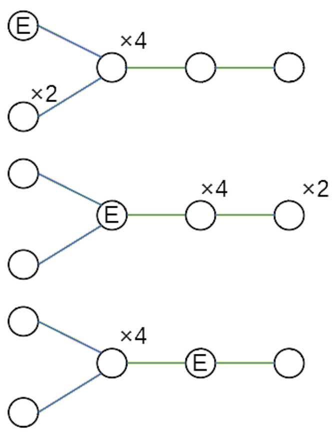


Figure 6. 20 trees with one degree = 3.

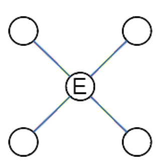


Figure 7. 1 tree with one degree = 4.

$$N = (8 + 8 + 8) + (8 + 8 + 4) + (1) = 45.$$

Question 2. Answer: `1,2,3,4,6,8,12,24`
In a full m-ary tree initially there is just one node (it is the root and also a leaf). After that the number of leaves can increase by $m - 1$ in a single step (one leaf becomes an internal node and creates exactly m children). The question now becomes – in how many arithmetic progressions both numbers 1 and 25 participate. We get these variants:

$$\begin{cases} 1, 2, 3, \ldots, 25 \\ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25 \\ 1, 4, 7, 10, 13, 16, 19, 22, 25 \\ 1, 5, 9, 13, 17, 21, 25 \\ 1, 7, 13, 19, 25 \\ 1, 9, 17, 25 \\ 1, 13, 25 \\ 1, 25 \end{cases}$$

In each variant we count the number of increments (how many times $m - 1$ was added in order to get 25). At every such step an inner node is created. The number of inner nodes can be computed as $\frac{25-1}{m-1}$.

m	Inner nodes
2	24
3	12
4	8
5	6
7	4
9	3
13	2
25	1

Question 3. Answer: `4,110`
Minimum height. There is one vertex (the root) at depth $d = 0$, at most 3 vertices at $d = 1$, at most 9 vertices at $d = 2$, at most 27 vertices at $d = 3$ and at most 81 vertices at $d = 4$. If maximum depth is 4 (i.e. the height of the tree is also 4), then there can be at most $1 + 3 + 9 + 27 + 81 = 121$ vertices in the tree. Which is just fine, since $111 < 121$.
Maximum height. In this case each vertex has just one child. The deepest node has depth $d = 110$ (one less than the number of nodes in the tree).

Question 4. Answer: `6`
We can sort cases depending on the location of the other vertex v_1 with degree 3 (it is known that another one is root; let us denote the root by v_0). That other vertex can have depth 1, 2 or 3.
Case 1. depth$(v_1) = 1$. There are three possible trees in this case; the two vertices of degree 2 can be either both children of v_0, or both of v_1, or one can be a child of v_1, and another of v_0.
Case 2. depth$(v_1) = 2$. There are two possible trees in this case; one of the vertices of degree 2 is beteween v_0 and v_1; but another vertex of degree 2 can be a child of v_0, or a child of v_1.
Case 3. depth$(v_1) = 3$. There is just one tree in this case. Both vertices of degree 2 are between v_0 and v_1.

Question 5. Answer: `2.125`
To geet the expected value of the random variable (the length of the code in bits) multiply the respective code lengths for all six letters of the alphabet by their respective probabilities:

$$1 \cdot \frac{1}{2} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} + 4 \cdot \frac{1}{16} + 4 \cdot \frac{1}{16} = 2\frac{1}{8}.$$

Note. Variable length codes where one code is never a prefix of another and all the codes can be arranged in a binary tree, where grouping them starts by the two least frequent letters (`!` and `c` in our case) is named *Huffman tree*.

Question 6. Answer: `C`
In any (infix, prefix or postfix) expression the number of binary operators (in our case the only such operator is $\to$) should be one less than the number of operands (in our case these are the Boolean variables a,b,c,d,e). Unary operators (in our case $\neg$) do not count.
In our expression so far there are 7 operators $\to$. There are also 9 Boolean variables. Therefore the triangle should become a binary operator $\to$. The expression looks like this:

$$\to \to \to \to \to \; a \; b \to \neg \, c \, \neg \, d \; c \; e \; \to \; \neg \, a \; \to \; d \; e.$$

Here is the regular (infix) representation of the same Boolean expression:

$$((((a \to b) \to (\neg c \to \neg d)) \to c) \to e) \to (\neg a \to (d \to e)).$$

Question 7. Answer: `17`

If we denote the rows/horizontals by numbers $1, 2, 3, 4$ and the columns/verticals by letters A, B, C, D as in ordinary chess, we can denote adding queens in the tree shown in Figure 8. The only two solutions are two shaded nodes at the bottom. All other placements of queens are dead ends.
Note. Not pursuing these dead ends (not going deeper, if we know that it is useless) reduces the number of nodes to consider to 17 (down from $4! = 24$ or even $4^4 = 256$, if we would apply *brute force*). This is not very much, but for larger chess boards the savings are larger. See https://bit.ly/3aQ1feo.

Question 8. Answer: `18,30,50`

You can determine the order of the vertices in the BFS traversal by counting vertices layer by layer (Figure 9).

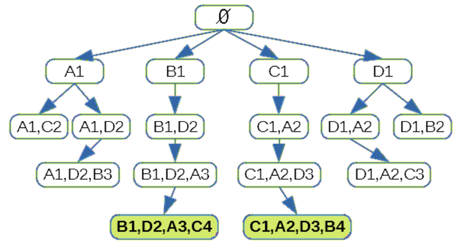


Figure 8. Backtracking tree: 4 Queens problem.

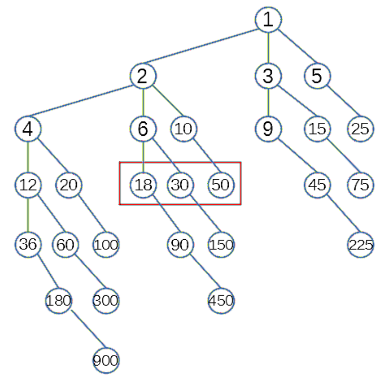


Figure 9. BFS tree with divisors of 900.

Note that the number $30 = \sqrt{900}$ is the 14th vertex in the BFS traversal (it is in the very middle).

Question 9. Answer: `37`

We add edges using Prim's algorithm:

Step 1	AB	4
Step 2	BC	8
Step 3	CI	2
Step 4	CF	4
Step 5	FG	2
Step 6	GH	1
Step 7	CD	7
Step 8	DE	9

Total weight:

$$4 + 8 + 2 + 4 + 2 + 1 + 7 + 9 = 37.$$

Question 10. Answer:

`(q (s (v (w w) v) s) (t (x (z z) x) (y y) t) q)  (r (u u) r)`

The DFS traversal creates two trees with q and r as their roots respectively.