Question 1: Computational Complexity (10 Points)

a) (2 Points)

Explain why **P** is closed under union and complement. Give brief algorithmic justifications for each.

b) (2 Points)

Let A be an arbitrary problem in **P**. Show that $A \in \mathbf{NP}$ and $A \in \mathbf{coNP}$ by giving separate justifications for each containment.

c) (2 Points)

Let A be the language of properly nested parentheses. For example, `(())` and `(()(()))()` are in A, but `)(` is not. Show that $A \in \mathbf{L}$.

d) (4 Points)

Consider the problem of deciding whether a propositional 3-SAT formula has at least two distinct satisfying assignments.

(i) Show this problem is in NP.

(ii) Show it is NP-hard by reduction from 3-SAT.

Hint: Given formula ϕ, create $\phi' = \phi \wedge (x \vee y \vee z)$ where x, y, z are new variables.

Question 2: A* Search, Randomized Algorithms **(10 Points)**

a) (4 Points)

Consider the A* search algorithm for finding a shortest path in a directed graph. Draw a graph with five nodes $\{s, x, a, b, t\}$ for which the algorithm **has to expand node x three times** when searching for the shortest path from s to t before the actual shortest path is found.

Provide weight values for the edges and heuristic values for each node in your sketch of the graph so that A* behaves in the intended way.

Illustrate how A* proceeds on your graph by listing the nodes in the order they are selected for expansion.

b) (2 Points)

Argue whether or not your heuristic function is admissible and/or monotonic.

c) (2 Points)

State a randomized algorithm that uses a fair coin to randomly select a letter from $\{A, B, C\}$ with equal probability for each letter. Argue the correctness of your algorithm.

d) (2 Points)

What is the expected number of coin flips your algorithm needs until a letter is selected?

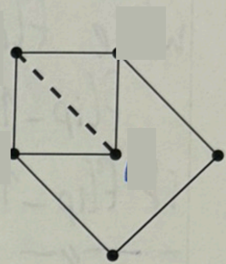
Question 3: Parameterized Algorithms (10 Points)

a) (6 Points)

Let C_4 be the cycle on four vertices. A graph contains C_4 as *induced subgraph* if it contains four vertices a, b, c, d, edges ab, bc, cd, da, and no further edges between these four vertices. Consider the following problem.

> C_4-FREE EDGE ADDITION
>
> *Instance*: A graph G and an integer k.
>
> *Parameter*: k.
>
> *Question*: Can one add at most k edges to G to ensure that the graph does not contain C_4 as induced subgraph?

For example, in the graph on the right, one can add the dashed edge to obtain the desired property.



Design an fpt algorithm for this problem, argue its correctness, and bound its run time in O-notation.

Hint: The problem is NP-complete. Algorithms with polynomial run time will be graded zero points.

b) (4 Points)

Recall that a reduction rule is *safe* if it does not change yes-instances to no-instances, or no-instances to yes-instances. Consider the following reduction rule for parameterized graph problems.

If the graph contains a leaf (a vertex of degree one), remove the leaf and its neighbor, and decrement k by one.

For each of the following four parameterized problems, mark whether the reduction rule is safe.

Grading: You obtain +1 point for each correct answer, −1 point for each incorrect answer, and 0 points for each question left unanswered. The total score for the subtask cannot be negative.

safe	not safe	problem
		INDEPENDENT SET
		DOMINATING SET
		VERTEX COVER
		CLIQUE