Fr. Conceicao Rodrigues College of Engineering
Fr. Agnel Ashram, Bandstand, Bandra (W), Mumbai - 400050

Department of Computer Engineering
Academic Term II: 23-24

Class: B.E (Computer), Sem – VI **Subject Name: Artificial Intelligence**

Student Name: **Roll No:**

Practical No:	**6**
Title:	Implementation of AO* algorithm
Date of Performance:	11-03-2024
Date of Submission:	18-03-2024

Rubrics for Evaluation:

Sr. No	**Performance Indicator**	**Excellent**	**Good**	**Below Average**	**Marks**
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Logic/Algorithm Complexity analysis (03)	03(Correct)	02(Partial)	01 (Tried)	
3	Coding Standards (03): Comments/indention/Naming conventions Test Cases /Output	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Assignment (03)	03(done well)	2 (Partially Correct)	1(submitted)	
Total					

Signature of the Teacher:

Experiment No: 6

Title: Implementation of AO* algorithm

Objective: To study AO* algorithm and implement it in an efficient manner

Theory:

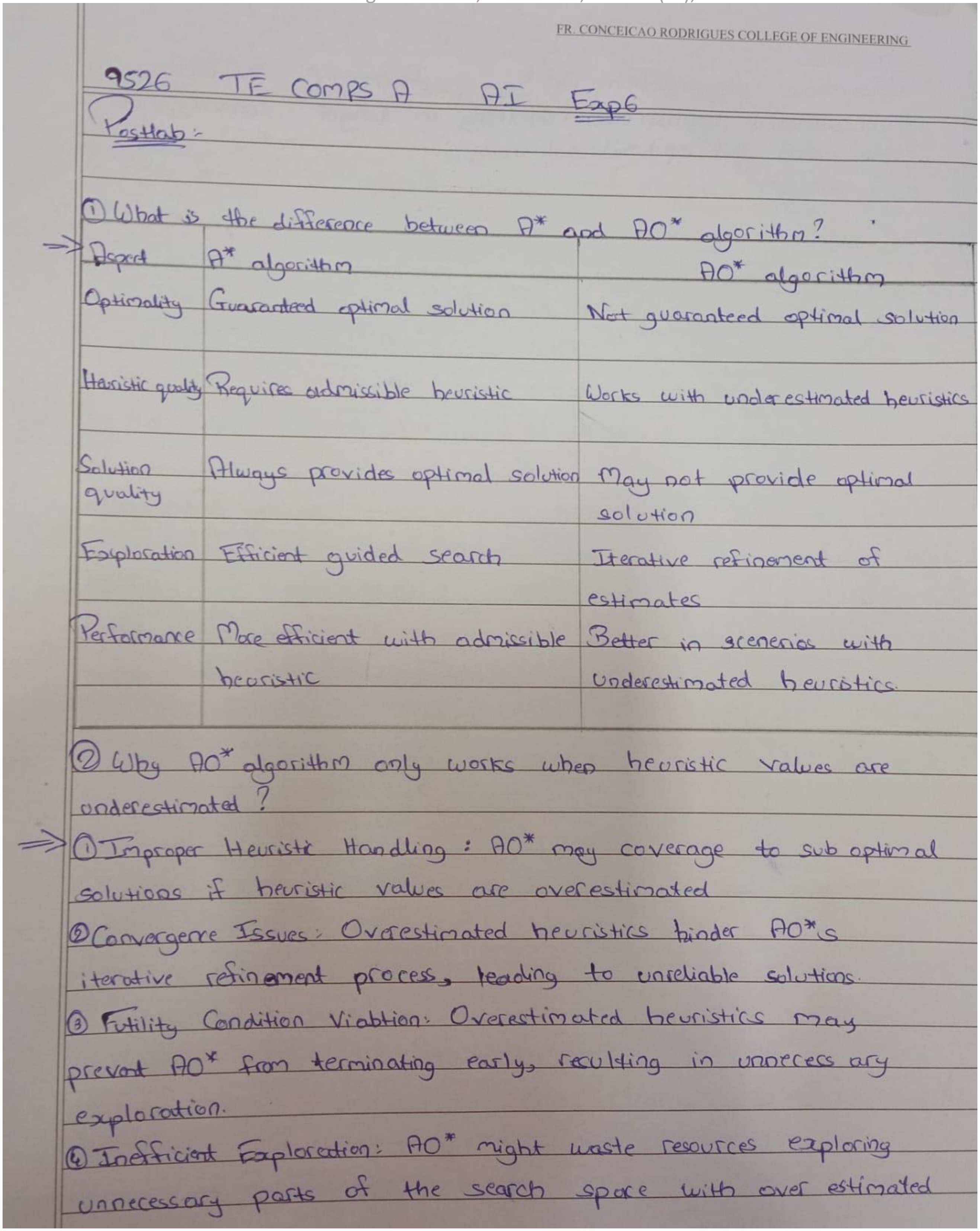
AO* Algorithm basically based on problem decomposition (Breakdown problem into small pieces). Basically, we will calculate the **cost function** here **(F(n)= G (n) + H (n))**
H: heuristic/ estimated value of the nodes. and **G:** actual cost or edge value (here unit value). Here we have taken the **edges value 1**, meaning we have to focus solely on the **heuristic value.**

Step-1: Create an initial graph with a single node (start node).
Step-2: Transverse the graph following the current path, accumulating node that has not yet been expanded or solved.
Step-3: Select any of these nodes and explore it. If it has no successors then call this value- FUTILITY else calculate f'(n) for each of the successors.
Step-4: If **f'(n)=0**, then mark the node as **SOLVED**.
Step-5: Change the value of f'(n) for the newly created node to reflect its successors by backpropagation.
Step-6: Whenever possible use the most promising routes, if a node is marked as SOLVED then mark the parent node as SOLVED.
Step-7: If the starting node is SOLVED or value is greater than **FUTILITY** then stop else repeat from Step-2.

OUTPUT:

Post Lab Assignment:

1. What is the difference between A* and AO* algorithm?
2. Why AO* algorithm only works when heuristic values are underestimated?



FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

9526 TE COMPS A AI Exp6

Postlab :-

① What is the difference between A* and AO* algorithm?

⇒

Aspect	A* algorithm	AO* algorithm
Optimality	Guaranteed optimal solution	Not guaranteed optimal solution
Heuristic quality	Requires admissible heuristic	Works with underestimated heuristics
Solution quality	Always provides optimal solution	May not provide optimal solution
Exploration	Efficient guided search	Iterative refinement of estimates
Performance	More efficient with admissible heuristic	Better in scenerios with underestimated heuristics.

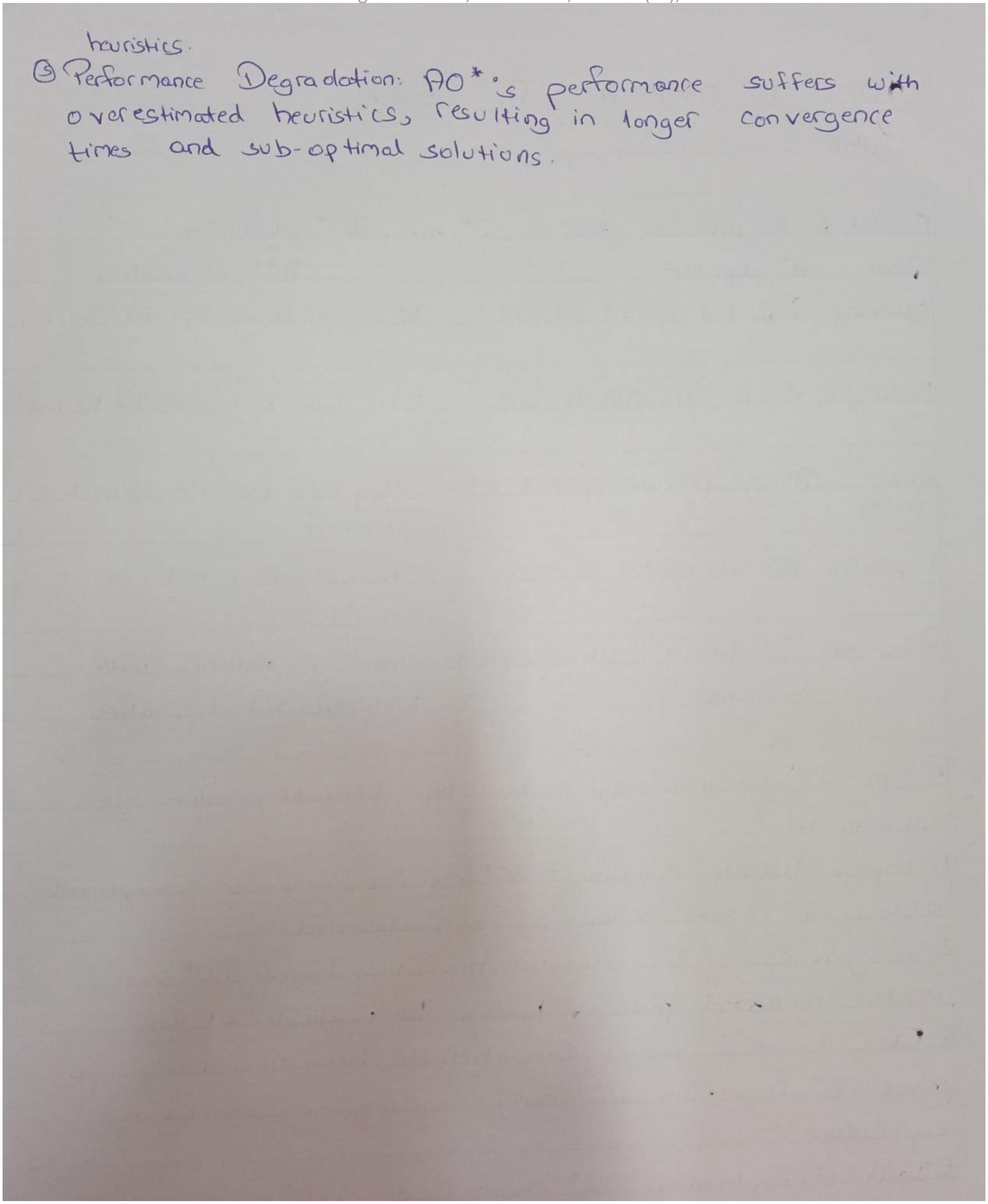
② Why AO* algorithm only works when heuristic values are underestimated?

⇒ ① Improper Heuristic Handling : AO* may coverage to sub optimal solutions if heuristic values are overestimated

② Convergence Issues: Overestimated heuristics hinder AO*s iterative refinement process, leading to unreliable solutions.

③ Futility Condition Viabtion: Overestimated heuristics may prevent AO* from terminating early, resulting in unnecessary exploration.

④ Inefficient Exploration: AO* might waste resources exploring unnecessary parts of the search space with over estimated



heuristics.

⑤ Performance Degradation: AO*'s performance suffers with overestimated heuristics, resulting in longer convergence times and sub-optimal solutions.