1. **Introduction**

   (a) **Turing Test**

   The so-called Turing test can be used to determine whether an artificial intelligence has a thinking ability that is equivalent to a human being. Give four necessary skills (in keywords) that an AI must have in order to have such a thinking ability. (2)

   1. Knowledge representation
   2. Making new conclusions
   3. Generating text content
   4. Understanding text

2. **Agents and Environment**

   (a) Different types of agents were presented in the lecture, including reflex agents and goal-based agents. What is the difference between these two types of agents? (2)

   al - based – ent make the actions through e goal. The decisions n be defferent for different enviroments!

   Reflex agent – has no state inside, chase the best action by the rules which are written for that enveroment choose wrong actions for another enviroment.

   (b) In the lecture, various environmental properties were used for classifying problems. State for 4 of these environmental properties the two (antagonistic) values. Also, provide an example of a suitable environment of an application for each the 4 characteristics and their 2 values (4)

   Single agent vs Multiagent →
   Deterministic vs Stohastic →
   Fully obserweable vs Partly observable →
   Static vs Dynamic →



CamScanner

3. **Search**

(a) **Search algorithm** The following algorithm is given in pseudocode:

```
search(node, goalTest):
        if (goalTest(node)) return node
        else for (child in node.children):
                childResult = search(child, goalTest)
                if (childResult is not failure) return childResult
        return failure
```

i. Which search algorithm is this? (1)

~~DFS~~

0/1

ii. Explain why this algorithm / implementation works well on trees, but may not work on graphs. (2)

~~This algorithm~~ works with recursion. First of all in all programming ~~languages the maximum deep~~ of the recursion is bordered.

0/2

~~Second the tree has the branching. But in the~~ so each ~~node has~~ it is parent. But in the graph it would be ~~it the recursion will be equal to number of nodes!~~

iii. If $b$ is the branching factor and $m$ the maximum depth of a **tree** (not graph), specify the memory and runtime complexity of the implementation/algorithm in Big-O notation. (2)

~~$O(b^m)$ - time complexity~~

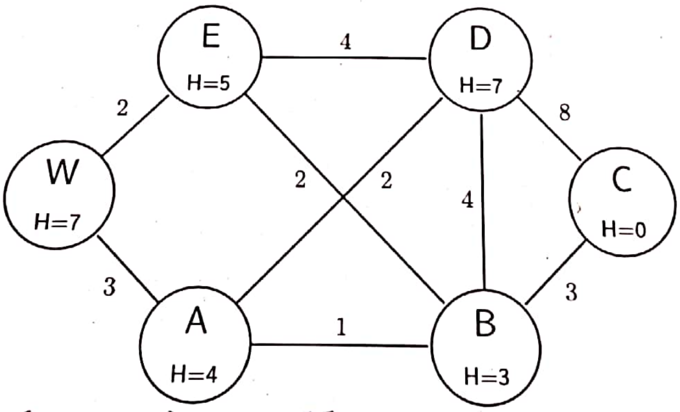
~~$O(b*m)$ - space complexity~~

0/2

(b) **A*** (3)

The following graph is given: The **heuristic (H)** is given in the respective node. The search should start in the node **W** and find the shortest path to the node **C**. All edges are bidirectional. Specify what condition must apply to the heuristic for the A* algorithm and state all nodes for which the heuristic is invalid.

0/3




~~W (3+4), (2+5)~~
~~(4+3), (5+7), (6+7)~~

4. **Complex Search**

(a) **Online / Offline Search** A robot agent is supposed to find the way out of a maze. There are 4 sensors available to him, which recognize whether there is a wall in front / left / right / behind

i. What environmental characteristic makes it necessary for an agent to use online search instead of offline search? (1)

~~It is possible that robot~~ will be in a cycle forever.
~~⇄~~. ~~In the offline way robot will be just reflex~~
~~agent~~. In the online way the robot will be ~~goal-based~~

(b) **Genetic Algorithms**

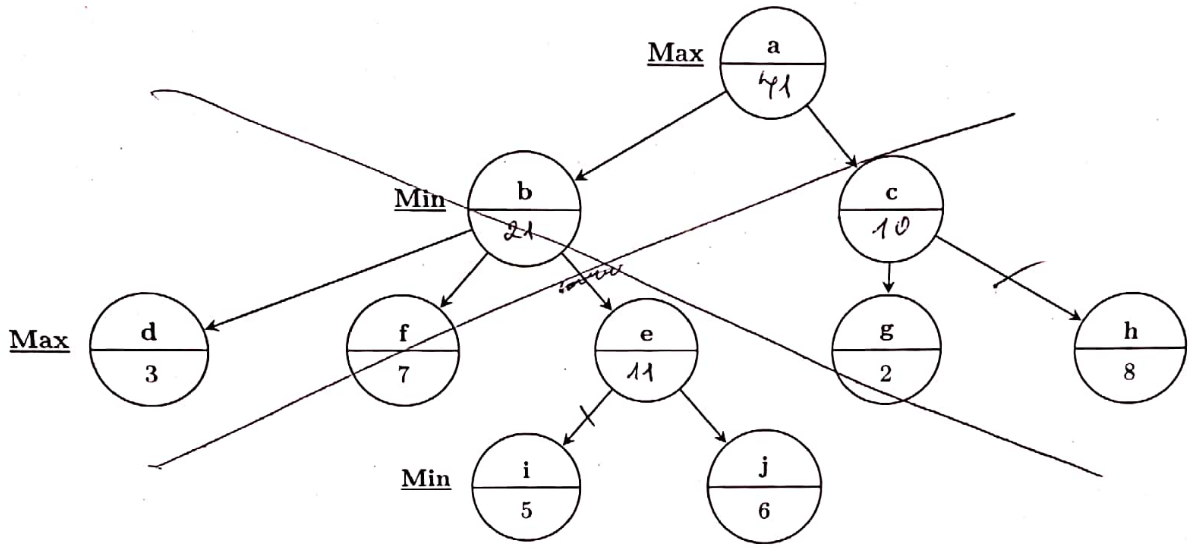
i. Name the four basic steps that make up genetic algorithms: (2)

~~1 generate~~
~~2 Mix~~
~~3 Test~~
~~4. Fix~~

ii. In practice, with genetic algorithms, the descendants are selected in a combination of fitness and chance. Why would it be bad to always select only the best children? (2)

5. **Antagonistic Search and Games**

(a) **Alpha/Beta Pruning**




i. Write in the bottom half of nodes a, b, c, and e the value that the node would have in a normal run of the Min-max algorithm in the end. (2)

ii. State all possible nodes that can be cut off by alpha / beta. Also indicate the type of cut. (3)

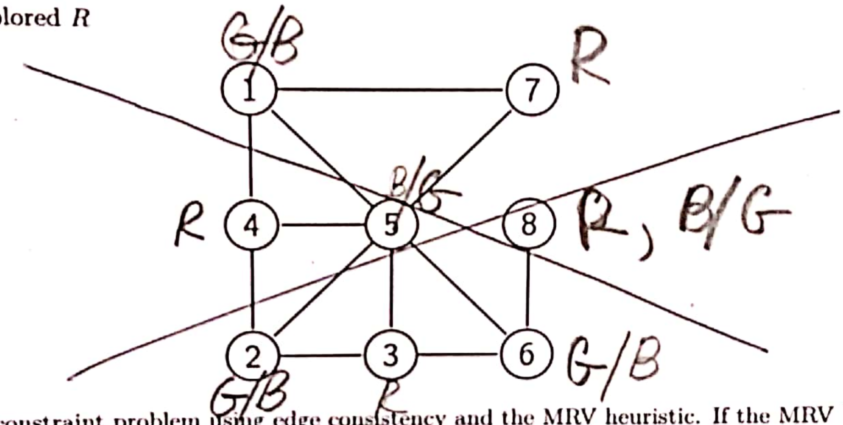
(b) ***Stochastic Games*** (5)

*State two fundamentally different techniques for stochastic games with random factors and explain which technique you would prefer.*

6. **Constraints**

In the following you should solve the graph coloring problem with the help of constraint solving methods. Each node in the graph below should be assigned a color. It should apply that:

1. Each node is assigned one of the colors $Red(R), Green(G), Blue(B)$
2. No two nodes connected by an edge are colored the same
3. Node 3 is colored $R$
4. Node 4 is colored $R$




(a) Solve the constraint problem using edge consistency and the MRV heuristic. If the MRV heuristic does not provide a unique variable, use the degree heuristic. If you have to choose between two colors for a node, choose preferably $R$, then $G$ then $B$. Enter all intermediate steps including the heuristic or consistency rule used in the table below. (6)

| Technic | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Node consistency | | | R | R | | | | |

7. **Logical Agents**

(a) **DPLL**

Gegeben sei folgende Klauselmenge $K$: (4)

1. $A \vee C \vee D$
2. $B \vee F \vee D$
3. $\neg D \vee \neg B$
4. $B \vee \neg C \vee A$
5. $B \vee F \vee C$
6. $\neg B \vee \neg F \vee \neg D$
7. $A \vee E$
8. $A \vee F$
9. $\neg F \vee C \vee \neg E$
10. $A \vee \neg C \vee \neg E$

Use the DPLL-Algorithm to show that the clause set $K$ is satisfiable. For each step state which clause(s) will then become true (The identifier of the clause(s) is enough). Note: You don't need to write down the simplified set of clauses.
If a step is not unique, select the variable in alphabetical order and set it to True first.

~~3 ∨ 2 → F~~

~~2 ∨ 6 → F~~

(b) **WalkSAT**

Explain the steps, how WalkSAT would find a model for a given set of clauses. (3)



CamScanner

8. **First order logic**

(a) Formulate the following statements in first-level predicate logic. Use only the specified predicates. (2)

i. Statement: If somebody is present and has been learning, than he passes the exam.
Predicates: $Present(x)$ (x is present), $PassesExam(x)$ (x passes the Exam), $HasLearned(x)$ (x has been learning for the exam)

~~$\exists x$ Present(x), HasLearned(x), PassedExam(x)~~

~~$\exists x$, Present(x) $\wedge$ {HasLearned(x) $\rightarrow$ PassesExam(x)~~

~~PassesExam (Present (HasLearned(x)))~~

ii. Statement: Every Student is enrolled in at least one course. (2)
Predicates: $Student(x)$ (x is a student), $Course(x)$ (x is a course), $enrolled(x, y)$ (x is enrolled ~~in y)~~

~~$\forall x$, Student(x) enrolled(Student(x), Course(x))~~

~~$\forall x$, enrolled Student(x), enrolled(x, course(x))~~

(b) **Unification** Specify a most common unifier for each of the following pairs of clauses. (3)

1. $King(x, England)$, $King(Wilhelm, y)$

~~$\exists x$, $\exists y$~~

2. $King(x)$, $King(x)$

~~$\forall x$~~

3. $Father(x, King(y))$, $Father(Robert, z)$

~~$\exists x$, $\exists y$, $\exists z$~~

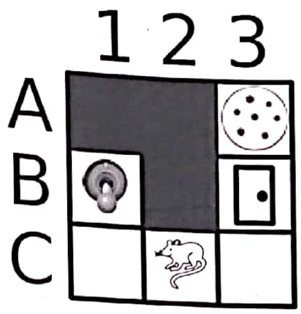
9. **Knowledge representation**

(a) **Ontologies**

Ontologies (used here as a generic term) can be divided into the following subclasses with regard to increasing power: catalog (glossary), classification (thesaurus) and semantic network (knowledge graph): How do these classes differ in terms of their expressiveness? (3)

10. **Planning**

A mouse (field C2) wants to reach a cookie (field A3), which is hidden behind a locked door (field B3), to eat it. The mouse can move one field at a time and only interact with objects (cookie/switch (field B1)) if it is on the same field. In order to go on the field with the door, the mouse must first flip the switch to open the door. If the mouse is on the field of the cookie, it can eat it. The black squares are walls and not valid fields.




(a) Model the problem as a planning problem in the **PDDL** language. For simplicity, you don't need to specify the initial state and the goal. Pay attention to the fact, that the mouse can only go from one square to a neighbouring square. (6)

Also specify the initial state, as well as the goal. You can assume, that a predicate $NextTo(x, y)$ exists, which for all fields x,y is true if and only if the two squares are next to each other horizontally or vertically.

**Hint:** Use, among others, a predicate On(x) to describe that the mouse is on field x, and a predicate to determine whether a field is walkable.