**Câu 1**

a. Biểu diễn tri thức là gì? Phương pháp biểu diễn tri thức, cho ví dụ:

Biểu diễn tri thức là cách mô tả cấu trúc và nội dung của tri thức trong máy tính hoặc hệ thống thông tin. Mục tiêu của việc biểu diễn tri thức là để máy tính có thể hiểu và xử lý thông tin một cách hiệu quả. Có nhiều phương pháp biểu diễn tri thức khác nhau, mỗi phương pháp thích hợp cho các loại tri thức và ứng dụng cụ thể.

Một trong những phương pháp biểu diễn tri thức phổ biến là biểu diễn dưới dạng Logic. Trong logic, tri thức được diễn đạt bằng các mệnh đề logic, điều này cho phép máy tính áp dụng các quy tắc logic để suy luận và xử lý tri thức.

Ví dụ: Phương pháp biểu diễn tri thức bằng Logic Mệnh đề, trong đó ta có thể sử dụng các biểu thức logic như "Nếu A thì B" hoặc "A nếu và chỉ nếu B". Ví dụ cụ thể:

- A: "Nắng lên"

- B: "Cỏ khô"

- Biểu diễn: "Nếu nắng lên thì cỏ khô."

b. 2.1. Biểu diễn chuẩn hội (CNF - Conjunctive Normal Form):

- CNF là một dạng biểu diễn tri thức trong logic mệnh đề.

- Các câu trong CNF được kết hợp bằng phép “và” (hội).

- Một câu trong CNF là một chuỗi các biến hoặc phủ định của biến đượckết hợp bằng phép “và”.

Ví dụ: Giả sử chúng ta có các biến A, B, C. Một số câu trong CNF có thể là:

- (A \land B)

- (\neg A \land B \land C)

- (A \land \neg B \land \neg C)

2.2. Biểu diễn chuẩn tuyển (DNF - Disjunctive Normal Form):

- DNF là một dạng biểu diễn tri thức trong logic mệnh đề.

- Các câu trong DNF được kết hợp bằng phép “hoặc” (tuyển).

- Một câu trong DNF là một chuỗi các biến hoặc phủ định của biến được kết hợp bằng phép “hoặc”.

Ví dụ: Giả sử chúng ta có các biến P, Q, R. Một số câu trong DNF có thể là:

- (P \lor Q)

- (P\neg P \lor Q \lor R)

- (P \lor \neg Q \lor \neg R)

2.3. Biểu diễn chuẩn Horn:

- Chuẩn Horn là một dạng đặc biệt của biểu diễn tri thức.

- Mỗi câu trong chuẩn Horn chỉ chứa tối đa một phủ định của biến.

- Các câu trong chuẩn Horn được kết hợp bằng phép “và” (hội).

Ví dụ: Câu trong chuẩn Horn có thể là:

- (P \land Q \Rightarrow R)

- (S \Rightarrow T)

**Câu 2**

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class GreedyColoring {

public static int[] greedyColoring(int[][] adjMatrix) {

int n = adjMatrix.length;

int[] colors = new int[n]; // Mảng màu, tất cả các thành phố chưa được tô màu

for (int city = 0; city < n; city++) {

Set<Integer> possibleColors = new HashSet<>(); // Tất cả các màu có thể sử dụng

for (int color = 1; color <= n; color++) {

possibleColors.add(color);

}

// Kiểm tra màu của các thành phố kề và loại bỏ các màu đã sử dụng

for (int neighbor = 0; neighbor < n; neighbor++) {

if (adjMatrix[city][neighbor] == 1 && possibleColors.contains(colors[neighbor])) {

possibleColors.remove(colors[neighbor]);

}

}

// Gán màu cho thành phố hiện tại

for (int color : possibleColors) {

colors[city] = color;

break; // Chọn một màu và thoát khỏi vòng lặp

}

}

return colors;

}

public static void main(String[] args) {

int[][] adjMatrix = {

{0, 1, 0, 1},

{1, 0, 1, 1},

{0, 1, 0, 1},

{1, 1, 1, 0}

};

int[] colors = greedyColoring(adjMatrix);

System.out.println("Màu của các thành phố: " + Arrays.toString(colors));

}

}

**Câu 3**

**3.2**

P(Cyes)=9/14; P(Cno) =5/14

Xác suất các thành phần

P(outlook=sunny|Cyes)=2/9

P(outlook=sunny|Cno)= 4/5

P(humidity=high|Cyes)=3/9

P(humidity=high|Cno)=4/5

P(wind=strong|Cyes)=3/9

P(wind=strong|Cno)=3/5

P(X|Cyes)= 2/9 \* 3/9 \* 3/9 =0.025

P(X|Cno)= 4/5 \* 4/5 \* 3/5 = 0.384

P(X|Cyes)\*P(Cyes)= 9/14 \* 0.025=0.02

P(X|Cno)\* P(Cno)= 5/14\* 0.384=0.14

X sẽ không chơi