

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: Khoa học máy tính TH TRÍ TUỆ NHÂN TẠO	BÀI 4 CÁC PHƯƠNG PHÁP GIẢI BÀI TOÁN THỎA MÃN RÀNG BUỘC	
--	---	--

A. MỤC TIÊU:

- Cung cấp kiến thức về các phương pháp giải bài toán thỏa mãn ràng buộc.
- Hiểu và sử dụng được các thuật toán vào trong phương pháp giải bài toán thỏa mãn ràng buộc.
- Cài đặt được các thuật toán theo phương pháp giải bài toán thỏa mãn ràng buộc.

B. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1. Kiến thức cần nhớ

- **Ràng buộc** là một quan hệ trên một tập các biến
- Ràng buộc có thể được biểu diễn bằng:
 - Một biểu thức (toán học/logic)
 - Một bảng liệt kê các phép gán giá trị phù hợp cho các biến
- **Bài toán thỏa mãn ràng buộc:**
 - Một tập hữu hạn các biến X
 - Miền giá trị cho mỗi biến D
 - Một tập hữu hạn các ràng buộc

Một lời giải thỏa mãn ràng buộc là một phép gán đầy đủ các giá trị của các biến sao cho thỏa mãn tất cả các ràng buộc.

- **Thuật toán tô màu tối ưu trên đồ thị**

Ràng buộc: 2 đỉnh kề nhau không tô cùng màu

Lặp lại các bước sau cho đến khi nào tô màu hết tất cả các đỉnh

Bước 1: Chọn đỉnh có bậc lớn nhất tô màu i .

Bước 2: Hạ bậc:

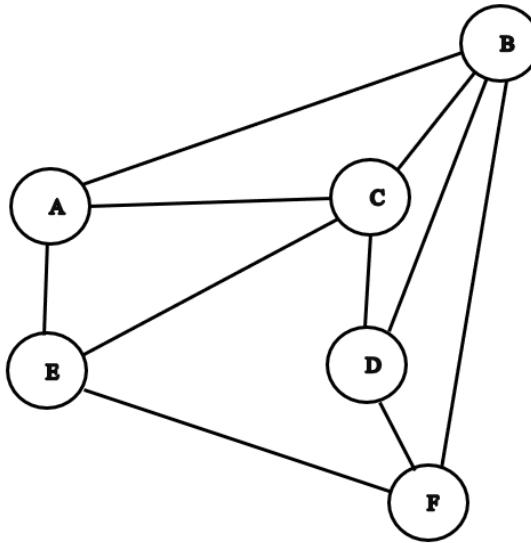
Đỉnh đã tô màu: $bậc = 0$

Những đỉnh có liên hệ: $bậc := bậc - 1$

Bước 3: Đánh dấu các đỉnh liên hệ và cấm tô màu i .

1.2. Giới thiệu bài tập mẫu

Bài mẫu. Cài đặt thuật toán tô màu đồ thị có 6 đỉnh, như sau:



Ma trận kề của đồ thị:

0	1	1	0	1	0
1	0	1	1	0	1
1	1	0	1	1	0
0	1	1	0	0	1
1	0	1	0	0	1
0	1	0	1	1	0

Hướng dẫn cài đặt:

```
# Ma trận kề của đồ thị 6 đỉnh
G = [[ 0, 1, 1, 0, 1, 0],
      [ 1, 0, 1, 1, 0, 1],
      [ 1, 1, 0, 1, 1, 0],
      [ 0, 1, 1, 0, 0, 1],
      [ 1, 0, 1, 0, 0, 1],
      [ 0, 1, 0, 1, 1, 0]]

# Tên các đỉnh của đồ thị.
node = "ABCDEF"
t_={}
for i in range(len(G)):
    t_[node[i]] = i

# Độ tuổi của các đỉnh
degree =[ ]
for i in range(len(G)):
    degree.append(sum(G[i]))
```

```

# Màu có thể sử dụng để tô cho các đỉnh
colorDict = {}
for i in range(len(G)):
    colorDict[node[i]] = ["Blue", "Red", "Yellow", "Green"]

# Sắp xếp các đỉnh theo thứ tự bậc
sortedNode = []
indeks = []

# use selection sort
for i in range(len(degree)):
    _max = 0
    j = 0
    for j in range(len(degree)):
        if j not in indeks:
            if degree[j] > _max:
                _max = degree[j]
                idx = j
    indeks.append(idx)
    sortedNode.append(node[idx])

# Phần xử lý màu (sử dụng màu ít nhất có thể)
theSolution = {}
for n in sortedNode:
    setTheColor = colorDict[n]
    theSolution[n] = setTheColor[0]
    adjacentNode = G[t_[n]]
    for j in range(len(adjacentNode)):
        if adjacentNode[j] == 1 and (setTheColor[0] in colorDict[node[j]]):
            colorDict[node[j]].remove(setTheColor[0])

# In kết quả từng đỉnh và màu đã tô tương ứng
for t, w in sorted(theSolution.items()):
    print("Đỉnh ", t, " = ", w)

```

2. Bài tập tại lớp

Bài 1: Phát triển code bài tập mẫu để đọc file ma trận kè dạng txt bất kỳ và in kết quả tô màu ra màn hình.

Bài 2: Phát triển code bài tập mẫu thành các chương trình con sao cho phù hợp.

Bài 3: Cài đặt thuật toán người bán hàng và ứng dụng tìm chu trình qua n thành phố mỗi thành phố qua 1 lần với chi phí tối thiểu.

3. Bài tập về nhà

Cài đặt giải thuật di truyền và ứng dụng vào bài toán sắp lịch dạy môn Toán và Tin học cho các giáo viên ở bậc PTTH trong 1 tuần. Các tham số khác có thể giả lập cho phù hợp với bài toán.