

# SUMMER CAMP 2022

## Trường Chuyên Phan Bội Châu – Nghệ An

presented by Đỗ Phan Thuận  
dophanthuan@gmail.com

Khoa Học Máy Tính  
Đại học Bách Khoa Hà Nội



Ngày 18 tháng 5 năm 2022



Bài 1. FACILITY

Bài 2. MEDPYRE

Bài 3. MEDPYRH

Bài 1. DELSEQ

Bài 2. CLOPAIR

Bài 3. ESEA

# FACILITY

Một công ty cung cấp dịch vụ cho thuê kho chứa hàng. Công ty nhận được  $n$  đơn đặt thuê kho hàng của khách hàng  $1, \dots, n$ , mỗi đơn thuê của khách hàng  $i$  sẽ bao gồm:

- ▶  $s_i$ : ngày bắt đầu thuê
- ▶  $d_i$ : số ngày cần thuê
- ▶  $r_i$ : số tiền khách hàng  $i$  thuê phải trả cho công ty

Tại mỗi thời điểm, kho hàng của công ty chỉ có thể phục vụ cho 1 đơn thuê duy nhất, đồng thời khi một khách hàng kết thúc sử dụng kho hàng thì công ty cần có  $K$  ngày để bảo trì kho trước khi cho một khách hàng khác thuê: cụ thể, khách thứ nhất kết thúc thuê vào ngày thứ  $x$  thì khách thứ hai chỉ có thể thuê sau ngày thứ  $x + K$ . Hãy giúp công ty lựa chọn các khách để cho thuê sao cho tổng số tiền thu được là lớn nhất.

## Thuật toán 30 điểm

Đúng với 30% số điểm có  $n, K \leq 10$ . Duyệt toàn bộ.

## Thuật toán 80 điểm

Đúng với  $n, K, s_i, d_i, r_i \leq 5 \times 10^4$

- ▶ Sort lại theo thời điểm kết thúc  $s$
- ▶ Quy hoạch động: gọi  $dp(i)$  là tổng số tiền thu được lớn nhất cho thuê đến ngày  $i$ :

$dp(i) = \text{MAX}(\$   
     $dp(i - 1)$  (nếu không cho thuê ngày  $i$ ),  
     $dp(j) + r(i)$  với  $j < i$ , nếu cho thuê ngày  $i$  và ngày gần nhất  
    cho thuê cách  $i$  ít nhất  $K$  ngày.

)

Độ phức tạp  $O(n^2)$ .

# Thuật toán 100 điểm

$$1 \leq n \leq 10^6, 1 \leq K \leq 10^9$$

Lưu ý: Với những dữ liệu vào có số lượng lớn ký tự ví dụ khoảng  $c \times 10^5$  đến  $c \times 10^6$  thì nên sử dụng phương pháp đọc ghi dữ liệu nhanh (FastI/O).

Do  $dp(i)$  tăng dần nên:

- ▶ Sử dụng phương pháp tìm kiếm nhị phân vị trí sát nhất có thời điểm kết thúc cách thời điểm bắt đầu của  $i$  ít nhất  $K$  ngày;
- ▶ hoặc BIT để tìm  $k$  tốt nhất trong đoạn trước  $i$ . Độ phức tạp  $O(n \log n)$ ;

## Phương pháp đọc ghi dữ liệu nhanh FastI/O

```
1
2
3 template <typename T> inline void read(T &x){
4     x = 0; char c;
5     while (!isdigit(c = getchar()));
6     do
7         x = x * 10 + c - '0';
8     while (isdigit(c = getchar()));
9 }
10
11 template <typename T> inline void write(T x){
12     if (x > 9) write(x / 10);
13     putchar(x % 10 + 48);
14 }
15
16 int main() {
17     read(n);
18     write(n);
19 }
```



Bài 1. FACILITY

Bài 2. MEDPYRE

Bài 3. MEDPYRH

Bài 1. DELSEQ

Bài 2. CLOPAIR

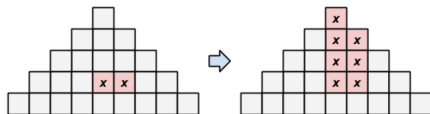
Bài 3. ESEA

## Kim tự tháp 1 — MEDPYRE

Khôi phục lại hoán vị tầng  $N$  của kim tự tháp khi biết giá trị  $x$  ở đỉnh. Mỗi giá trị trong 1 ô của kim tự tháp bằng trung vị của 3 số liền dưới.

100%

- ▶  $x = 1$  hoặc  $2N - 1$ , câu trả lời là “No”. Tất cả các giá trị khác của  $x$  đều cho câu trả lời “Yes”
- ▶ Nhận xét: nếu trên hàng  $i$  có 2 ô liên tiếp  $j$  và  $j + 1$  có cùng giá trị  $x$  thì tất cả các ô thứ  $j$  và  $j + 1$  (nếu tồn tại) ở tất cả các hàng nhỏ hơn  $i$  cũng có giá trị  $x$
- ▶ Vì vậy, chỉ cần đưa ra một hoán vị để tạo ra hai ô liên tiếp thứ  $N$  và  $N + 1$  là  $x$  trên hàng thứ  $N - 1$ . Ví dụ:
  - ▶ nếu  $x \neq 2$ , một kết quả có thể là  $(\dots, x - 1, x, x + 1, x - 2, \dots)$
  - ▶ nếu  $x = 2$ , một kết quả có thể là  $(\dots, x + 1, x, x - 1, x + 2, \dots)$



Bài 1. FACILITY

Bài 2. MEDPYRE

**Bài 3. MEDPYRH**

Bài 1. DELSEQ

Bài 2. CLOPAIR

Bài 3. ESEA

## Kim tự tháp 2 — MEDPYRH

Khôi phục giá trị  $x$  ở đỉnh kim tự tháp khi biết hoán vị tầng  $N$ . Mỗi giá trị trong 1 ô của kim tự tháp bằng trung vị của 3 số liền dưới.

Subtask 1:  $\sum N \leq 5000$

Thuật toán trực tiếp cho độ phức tạp  $N^2$



## Subtask 3: $N \leq 10^5$

Tìm kiếm nhị phân:

- ▶ Tìm kiếm nhị phân giá trị  $x$  sử dụng  $O(\log Q)$  câu hỏi, với  $Q$  là giá trị lớn nhất có thể, và trả lời câu hỏi: Giá trị ô trên đỉnh kim tự tháp ít nhất là  $x$  hay không?
- ▶ Với mỗi ô, cần quan tâm giá trị trong ô có ít nhất là  $x$  hay không
- ▶ thay thế tất cả các số nguyên nhỏ hơn  $x$  bằng 0 và các số nguyên khác bằng 1. Bài toán đưa về subtask 2
- ▶ DPT:  $O(N \log Q)$



Bài 1. FACILITY

Bài 2. MEDPYRE

Bài 3. MEDPYRH

**Bài 1. DELSEQ**

Bài 2. CLOPAIR

Bài 3. ESEA

## Xoá chuỗi — DELSEQ

Thực hiện nhiều lần phép xoá  $2^i$  ký tự liên tiếp trên chuỗi  $A$  để xác định giá trị lớn nhất có thể của chuỗi, độ dài của các chuỗi đã xoá phải đôi một khác nhau

- ▶ Với độ dài  $N$ , số lượng nhiều nhất các phép toán có thể được thực hiện là  $\lfloor \log_2(N) \rfloor + 1$ .
- ▶ Gọi  $F_{i,S}$  là giá trị tổng lớn nhất có thể tạo được bởi một dãy con kết thúc tại vị trí  $i$ , nếu các phép toán được thực hiện trong tập  $S$ . Ta có công thức:

$$F_{i,S} = \max\{F_{i-1,S} + A_i, F_{i-2^k,S-\{k\}} \mid k \in S\}$$

- ▶ Có 2 trường hợp trong công thức:
  1. Không xóa các phần tử khỏi vị trí  $i$  và thêm  $A_i$  vào tổng,
  2. Xóa các phần tử khỏi vị trí  $i$  với độ dài  $2^k$  bởi một phép toán có trong tập  $S$
- ▶ ĐPT:  $O(N^2 \log N)$

Bài 1. FACILITY

Bài 2. MEDPYRE

Bài 3. MEDPYRH

Bài 1. DELSEQ

**Bài 2. CLOPAIR**

Bài 3. ESEA

# CLOPAIR

Cho  $N$  điểm trên mặt phẳng, hãy tìm một cặp điểm với khoảng cách ơclit nhỏ nhất giữa chúng. Biết rằng không có hai điểm nào trùng nhau và có duy nhất một cặp có khoảng cách nhỏ nhất.

## Subtask

Sử dụng 2 vòng lặp duyệt qua tất cả các cặp điểm cho độ phức tạp  $O(n^2)$ .

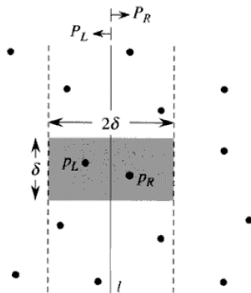
# Thuật toán 100 điểm

$$2 \leq N \leq 50000$$

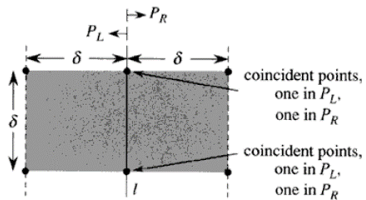
Sử dụng phương pháp chia để trị. Tại mỗi thủ tục đệ qui đầu vào là tập điểm cần tìm cặp điểm gần nhất:

- ▶ CHIA : chia tập điểm thành 2 tập đều nhau (theo toạ độ  $x$ , mặt phẳng được chia thành 2 nửa).
- ▶ XỬ LÝ: gọi đệ qui cho mỗi tập vừa chia.
- ▶ KẾT HỢP: thuật toán đưa ra cặp điểm gần nhất.

# Thuật toán 100 điểm<sup>2</sup>



(a)



(b)



## Thuật toán 100 điểm

- ▶ CHIA: chia tập điểm thành 2 tập đều nhau (theo toạ độ  $x$ , mặt phẳng được chia thành 2 nửa):  $P_L, P_R$  (ĐPT  $O(n)$ ).
- ▶ XỬ LÝ: gọi đệ qui cho mỗi tập vừa chia. Kết quả thu được 2 khoảng cách của các cặp điểm gần nhất bên nửa trái và nửa phải:  $(\delta_L, \delta_R)$ .
- ▶ KẾT HỢP:
  - ▶ lấy giá trị nhỏ nhất của 2 khoảng cách thu được ở trên:  
 $\delta = \min\{\delta_L, \delta_R\}$  (độ phức tạp  $O(1)$ ).
  - ▶ kiểm tra trong các cặp điểm mà một điểm  $\in P_L$  và một điểm  $\in P_R$  có khoảng cách nhỏ hơn  $\delta$  không:
    - ▶ Tạo một mảng  $Y'$  các điểm nằm trong giới hạn  $2\delta$  theo trục dọc ở giữa trục chia, sắp xếp  $Y'$  tăng dần theo toạ độ  $y$  (độ phức tạp  $O(n \log n)$  hoặc  $O(n)$ ).
    - ▶ Với mỗi điểm trong  $Y'$ , so sánh điểm đó với 7 điểm tiếp theo (độ phức tạp là  $O(7n)$ ). Nếu tìm được khoảng cách  $\delta' < \delta$ , cập nhật lại  $\delta$  theo  $\delta'$ .

$$T(n) = O(1), \text{ nếu } n \leq 3.$$

$$T(n) = 2T(n/2) + O(n \log n).$$

Độ phức tạp tổng cộng là  $O(n \log^2 n)$ .

## Các cải tiến

- ▶ Tiền xử lý:  $X$  là tập điểm đã sắp xếp toàn bộ các điểm theo toạ độ  $x$ . Khi đó mỗi lần chia một nửa tập theo toạ độ  $x$  thì vẫn giữ được các điểm sắp xếp tăng dần theo  $x$  ở mỗi tập.
- ▶ Tiền xử lý:  $Y$  là tập điểm đã sắp xếp toàn bộ các điểm theo toạ độ  $x$ . Điều này giúp cho xác định được tập điểm  $Y'$  trong mỗi thủ tục đệ qui sắp xếp theo toạ độ  $y$  với độ phức tạp  $O(n)$ .
- ▶ Độ phức tạp  $O(n \log n)$ :  
$$T(n) = O(1), \text{ nếu } n \leq 3.$$
$$T(n) = 2T(n/2) + O(n).$$
- ▶ Chỉ cần so sánh với 5 điểm tiếp theo thay vì 6 điểm.
- ▶ Mở rộng sang bài toán 3 chiều.

Bài 1. FACILITY

Bài 2. MEDPYRE

Bài 3. MEDPYRH

Bài 1. DELSEQ

Bài 2. CLOPAIR

Bài 3. ESEA

## Bày trận biển Đông — ESEA

Phân hoạch tập các cặp bộ dò ra thành ít nhất các nhóm sao cho mỗi cặp phải thuộc vào đúng một nhóm và trong mỗi nhóm không có hai cặp nào có liên kết vẹn toàn với nhau.

## Đưa về bài toán đồ thị

- ▶ Mỗi cặp bộ dò là một đỉnh của đồ thị
- ▶ Hai cặp bộ dò  $i$  và  $j$  tạo thành một cạnh nếu như chúng có liên kết vẹn toàn với nhau
- ▶ Bài toán đưa về tìm tập độc lập có lực lượng lớn nhất
- ▶ Đồ thị ở đây có tính chất đặc biệt “comparability”, nghĩa là  $a \prec b, b \prec c$  thì  $a \prec c$
- ▶ Bài toán dãy con tăng dài nhất chính là bài toán tập độc lập cực đại trên đồ thị mà mỗi đỉnh tương ứng với 1 số, mỗi cung tương ứng với số bên trái nhỏ hơn số bên phải. Ví dụ dãy: 5 3 7 thì đồ thị tương ứng có 3 đỉnh và 2 cung 57 và 37

## Subtask 1

$n \leq 10$  : Duyệt mọi tổ hợp.

## Subtask 2

- ▶  $n \leq 1000$ ; và trong tất cả các cặp, hai bộ dò của chúng đặt tại cùng một điểm, nghĩa là  $x_1 = x_2, y_1 = y_2$
- ▶ Cặp bộ dò suy biến về thành 1 điểm
- ▶ Giải bài toán tập độc lập trên đồ thị hoán vị giống như bài toán tìm dãy con dài nhất
- ▶ Thuật toán quy hoạch động đơn giản cho độ phức tạp  $O(n^2)$ .

## Subtask 3

- ▶  $n \leq 10^6$ ; và trong tất cả các cặp, hai bộ dò của chúng đặt tại cùng một điểm, nghĩa là  $x_1 = x_2, y_1 = y_2$
- ▶ Cặp bộ dò suy biến về thành 1 điểm
- ▶ Giải bài toán tập độc lập trên đồ thị hoán vị giống như bài toán tìm dãy con dài nhất
- ▶ Thuật toán quy hoạch động kết hợp với chia để trị hoặc cấu trúc dữ liệu BIT cho độ phức tạp  $O(n \log n)$ .



## Subtask 4

- ▶  $n \leq 1000$
- ▶ Giải bài toán tập độc lập trên đồ thị hình thang giống như bài toán tìm dãy con dài nhất
- ▶ Thuật toán quy hoạch động đơn giản cho độ phức tạp  $O(n^2)$ .

## Subtask 5

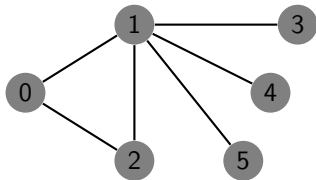
- ▶  $n \leq 10^6$
- ▶ Giải bài toán tập độc lập trên đồ thị hình thang giống như bài toán tìm dãy con dài nhất
- ▶ Thuật toán quy hoạch động kết hợp với chia để trị hoặc cấu trúc dữ liệu BIT cho độ phức tạp  $O(n \log n)$
- ▶ Do một hình thang được đại diện bởi hai cặp  $(x_1, y_1), (x_2, y_2)$  nên cần phải sắp xếp theo thứ tự  $x_1, y_1, x_2, y_2$  để ưu tiên theo thứ tự cặp  $(x_1, y_1)$  và truy vấn theo BIT hoặc chia nhị phân theo cặp còn lại  $(x_2, y_2)$
- ▶ Truy vết: các bộ dò cùng mức trong quá trình quy hoạch động sẽ cùng một nhóm
- ▶ Có thể áp dụng cây BIT2D cho hai cặp tọa độ của hình thang cho độ phức tạp  $O(n \log^2 n)$ . Có thể mất một số test quá thời gian.

## Một số bài toán quen thuộc trên đồ thị

- ▶ Đây là những bài toán cơ bản rất hay được sử dụng để ra bài trong các kỳ thi
- ▶ Thường xuyên được ẩn chứa kín trong phát biểu bài toán
- ▶ Xét một số ví dụ

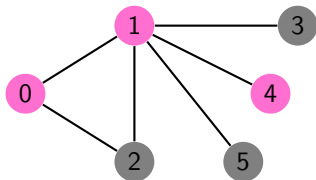
## Bài toán phủ đỉnh - MVC

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một phủ đỉnh là một tập con các đỉnh  $S$ , sao cho với mỗi cạnh  $(u, v) \in E$ , hoặc  $u$  hoặc  $v$  (hoặc cả hai) thuộc  $S$



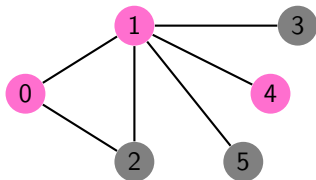
## Bài toán phủ đỉnh - MVC

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một phủ đỉnh là một tập con các đỉnh  $S$ , sao cho với mỗi cạnh  $(u, v) \in E$ , hoặc  $u$  hoặc  $v$  (hoặc cả hai) thuộc  $S$



## Bài toán phủ đỉnh - MVC

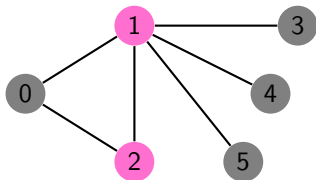
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một phủ đỉnh là một tập con các đỉnh  $S$ , sao cho với mỗi cạnh  $(u, v) \in E$ , hoặc  $u$  hoặc  $v$  (hoặc cả hai) thuộc  $S$



- ▶ Hãy tìm một phủ đỉnh có lực lượng nhỏ nhất

## Bài toán phủ đỉnh - MVC

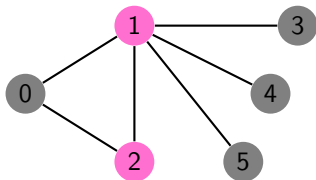
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một phủ đỉnh là một tập con các đỉnh  $S$ , sao cho với mỗi cạnh  $(u, v) \in E$ , hoặc  $u$  hoặc  $v$  (hoặc cả hai) thuộc  $S$



- ▶ Hãy tìm một phủ đỉnh có lực lượng nhỏ nhất

## Bài toán phủ đỉnh - MVC

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một phủ đỉnh là một tập con các đỉnh  $S$ , sao cho với mỗi cạnh  $(u, v) \in E$ , hoặc  $u$  hoặc  $v$  (hoặc cả hai) thuộc  $S$

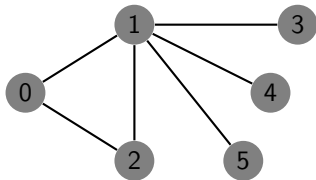


- ▶ Hãy tìm một phủ đỉnh có lực lượng nhỏ nhất
- ▶ Đây là bài toán NP-khó đối với đồ thị tổng quát



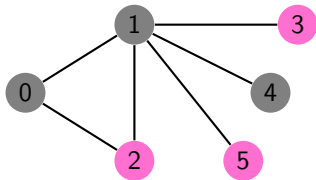
## Bài toán tập độc lập - MIS

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tập độc lập là một tập con các đỉnh  $S$ , sao cho không có hai đỉnh  $u, v$  nào trong  $S$  kề với nhau trong  $G$



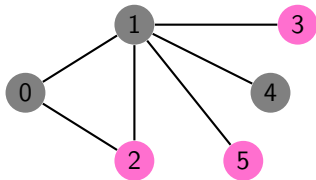
## Bài toán tập độc lập - MIS

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tập độc lập là một tập con các đỉnh  $S$ , sao cho không có hai đỉnh  $u, v$  nào trong  $S$  kề với nhau trong  $G$



## Bài toán tập độc lập - MIS

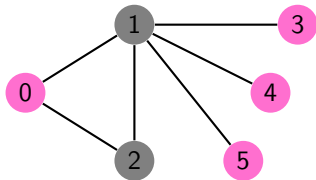
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tập độc lập là một tập con các đỉnh  $S$ , sao cho không có hai đỉnh  $u, v$  nào trong  $S$  kề với nhau trong  $G$



- ▶ Hãy tìm một tập độc lập có lực lượng lớn nhất

## Bài toán tập độc lập - MIS

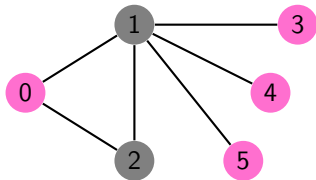
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tập độc lập là một tập con các đỉnh  $S$ , sao cho không có hai đỉnh  $u, v$  nào trong  $S$  kề với nhau trong  $G$



- ▶ Hãy tìm một tập độc lập có lực lượng lớn nhất

## Bài toán tập độc lập - MIS

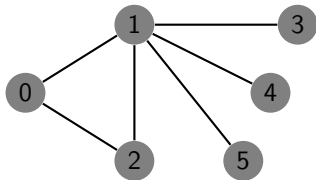
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tập độc lập là một tập con các đỉnh  $S$ , sao cho không có hai đỉnh  $u, v$  nào trong  $S$  kề với nhau trong  $G$



- ▶ Hãy tìm một tập độc lập có lực lượng lớn nhất
- ▶ Đây là bài toán NP-khó đối với đồ thị tổng quát

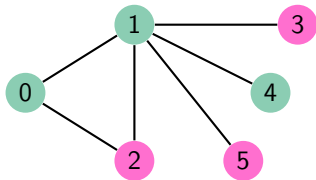
## Mối quan hệ giữa MVC và MIS

- ▶ Hai bài toán trên có mối liên quan chặt chẽ với nhau
- ▶ Một tập con của các đỉnh là một phủ đỉnh khi và chỉ khi tập bù của nó là tập độc lập



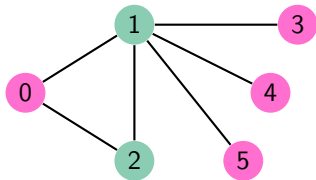
## Mối quan hệ giữa MVC và MIS

- ▶ Hai bài toán trên có mối liên quan chặt chẽ với nhau
- ▶ Một tập con của các đỉnh là một phủ đỉnh khi và chỉ khi tập bù của nó là tập độc lập



## Mối quan hệ giữa MVC và MIS

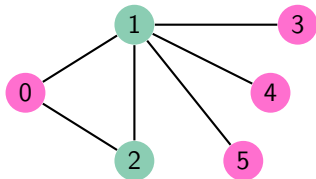
- ▶ Hai bài toán trên có mối liên quan chặt chẽ với nhau
- ▶ Một tập con của các đỉnh là một phủ đỉnh khi và chỉ khi tập bù của nó là tập độc lập





## Mối quan hệ giữa MVC và MIS

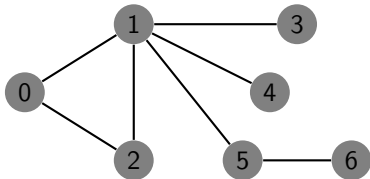
- ▶ Hai bài toán trên có mối liên quan chặt chẽ với nhau
- ▶ Một tập con của các đỉnh là một phủ đỉnh khi và chỉ khi tập bù của nó là tập độc lập



- ▶ Lực lượng của một phủ tập có kích thước nhỏ nhất cộng với lực lượng của tập độc lập có kích thước lớn nhất bằng tổng số đỉnh của đồ thị

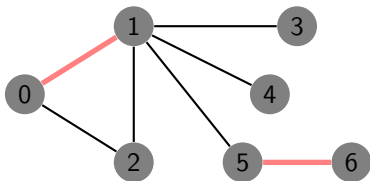
## Bài toán ghép cặp

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một cặp ghép là một tập con các cạnh  $F$  sao cho mỗi đỉnh kề với tối đa một cạnh trong  $F$



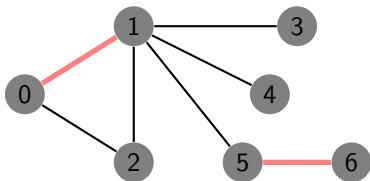
## Bài toán ghép cặp

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một cặp ghép là một tập con các cạnh  $F$  sao cho mỗi đỉnh kề với tối đa một cạnh trong  $F$



## Bài toán ghép cặp

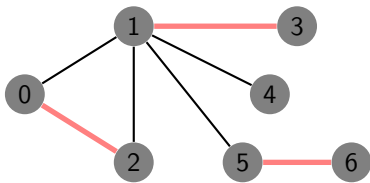
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một cặp ghép là một tập con các cạnh  $F$  sao cho mỗi đỉnh kề với tối đa một cạnh trong  $F$



- ▶ Hãy tìm một cặp ghép có lực lượng lớn nhất

## Bài toán ghép cặp

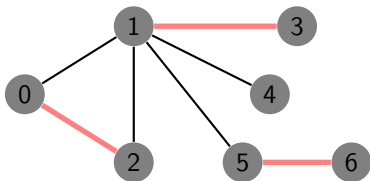
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một cặp ghép là một tập con các cạnh  $F$  sao cho mỗi đỉnh kề với tối đa một cạnh trong  $F$



- ▶ Hãy tìm một cặp ghép có lực lượng lớn nhất

## Bài toán ghép cặp

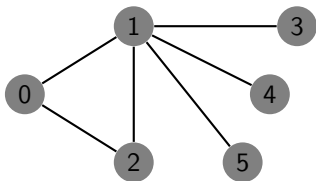
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một cặp ghép là một tập con các cạnh  $F$  sao cho mỗi đỉnh kề với tối đa một cạnh trong  $F$



- ▶ Hãy tìm một cặp ghép có lực lượng lớn nhất
- ▶ Có thuật toán  $O(|V|^4)$  cho đồ thị tổng quát, nhưng cài đặt khá phức tạp

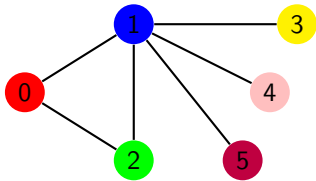
## Bài toán tô màu đồ thị

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tô màu của đồ thị là một cách gán các màu vào các đỉnh sao cho các đỉnh kề nhau không cùng màu



## Bài toán tô màu đồ thị

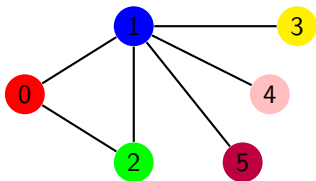
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tô màu của đồ thị là một cách gán các màu vào các đỉnh sao cho các đỉnh kề nhau không cùng màu





## Bài toán tô màu đồ thị

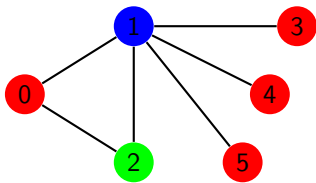
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tô màu của đồ thị là một cách gán các màu vào các đỉnh sao cho các đỉnh kề nhau không cùng màu



- ▶ Hãy tìm một cách tô màu sao cho sử dụng ít màu khác nhau nhất

## Bài toán tô màu đồ thị

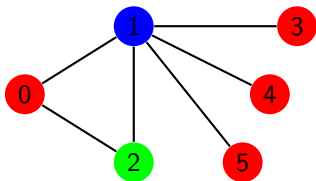
- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tô màu của đồ thị là một cách gán các màu vào các đỉnh sao cho các đỉnh kề nhau không cùng màu



- ▶ Hãy tìm một cách tô màu sao cho sử dụng ít màu khác nhau nhất

## Bài toán tô màu đồ thị

- ▶ Cho đồ thị vô hướng không trọng số  $G = (V, E)$
- ▶ Một tô màu của đồ thị là một cách gán các màu vào các đỉnh sao cho các đỉnh kề nhau không cùng màu



- ▶ Hãy tìm một cách tô màu sao cho sử dụng ít màu khác nhau nhất
- ▶ Đây là bài toán NP-khó trên đồ thị tổng quát