Tên bài giảng

Phương pháp phân tích thiết kế tham lam

Môn học: **Phân tích thiết kế thuật toán**

Chương: 3

Hệ: Đại học

Giảng viên: TS. Phạm Đình Phong

Email: phongpd@utc.edu.vn

Nội dung bài học

1. **Phân tích thiết kế, đánh giá thuật toán tham lam**
2. **Một số thuật toán tham lam theo nguyên lý thứ tự**
3. **Một số thuật toán xấp xỉ**

Phương pháp tham lam

- Mục đích
  - Tìm một lời giải tốt trong thời gian chấp nhận được (độ phức tạp đa thức)
- Ý tưởng
  - Dựa vào sự đánh giá tối ưu cục bộ địa phương (local optimum) để đưa ra quyết định tức thì tại mỗi bước lựa chọn, với hy vọng cuối cùng sẽ tìm ra được phương án tối ưu toàn cục (global optimum).

Phương pháp tham lam

- Ví dụ
  - Giả sử có n loại tiền giấy, loại tiền thứ i có mệnh giá là v_i (đồng). Hãy chỉ ra cách trả dùng ít tờ tiền nhất để mua một mặt hàng có giá là m đồng

Phương pháp tham lam

- Ví dụ
  - Bài toán này là một bài toán có thể giải theo nhiều cách
    - Nếu có các loại tiền như hệ thống tiền tệ của Việt Nam: 1k đồng, 2k đồng, 5k đồng, 10k đồng, 20k đồng, 50k đồng, 100k đồng, 200k đồng, 500k đồng thì để trả món hàng giá 9k đồng
      - Một người máy được cài đặt chiến lược tìm kiếm vét cạn sẽ duyệt tổ hợp của tất cả các tờ tiền và tìm tổ hợp gồm ít tờ tiền nhất có tổng mệnh giá là 9k đồng. Vấn đề sẽ xảy ra nếu món hàng không phải có giá 9k đồng mà là 9k tỉ đồng thì người máy sẽ nghĩ đến khi hết điện thì thôi

Phương pháp tham lam

- Ví dụ
  - Bài toán này là một bài toán có thể giải theo nhiều cách
    - Một người máy khác được cài đặt chiến lược chia để trị
    - Để trả 9k đồng dĩ nhiên không cần dùng đến những tờ tiền mệnh giá lớn hơn 9k. Bài toán trở thành đổi 9k đồng ra những tờ tiền 1k đồng, 2k đồng và 5k đồng. Nếu có một tờ x đồng trong phương án tối ưu thì vấn đề còn lại là đổi $9k - x$ đồng, quy về ba bài toán con (với $x = 1k, 2k, 5k$), giải chúng và chọn phương án tốt nhất trong ba lời giải
    - Khi mà số tiền cần quy đổi rất lớn, giới hạn về thời gian và bộ nhớ sẽ làm cho giải pháp của người máy này bất khả thi

Phương pháp tham lam

- Ví dụ
  - Bài toán này là một bài toán có thể giải theo nhiều cách
    - Bây giờ hãy thử nghĩ theo một cách rất con người, khi các bà nội trợ đi mua sắm, họ chẳng có máy tính, không có khái niệm gì về thuật toán chia để trị, quy hoạch động, vét cạn,… (mà có biết họ cũng chẳng dùng).
    - Để trả 9k đồng, chắc chắn họ sẽ trả bằng 1 tờ 5k đồng và 2 tờ 2k đồng. Cách làm của họ rất đơn giản, mỗi khi rút một tờ tiền ra trả, họ quyết định tức thời bằng cách lấy ngay tờ tiền mệnh giá cao nhất không vượt quá giá trị cần trả, mà không cần để ý đến “hậu quả” của sự quyết định đó.

Phương pháp tham lam

- Kỹ thuật
  - Sắp xếp các lựa chọn cho bước đó theo thứ tự nào đó "có lợi" (tăng dần hoặc giảm dần)
  - Chọn lựa chọn tốt nhất rồi đi tiếp bước kế (không quay lui)

  → Có thể áp dụng hàng đợi ưu tiên
- Đặc điểm
  - Có trường hợp luôn tìm ra phương án tối ưu
  - Trường hợp không tìm ra phương án tối ưu thì thu được một phương án khả dĩ chấp nhận được
  - Thường có tốc độ tốt hơn hẳn so với các thuật toán tối ưu toàn cục.

Phương pháp tham lam

- Về lý thuyết, có thể chỉ ra những hệ thống tiền tệ mà cách làm này cho ra giải pháp không tối ưu
  - Giả sử có 3 loại tiền: 1k đồng, 5k đồng và 8k đồng. Nếu cần đổi **10k** đồng thì phương án tối ưu phải là dùng 2 tờ 5k đồng nhưng phương pháp tham lam sẽ cho phương án 1 tờ 8k đồng và 2 tờ 1k đồng. Hậu quả của phép chọn tờ tiền 8k đồng đã làm cho giải pháp cuối cùng không tối ưu. Dù sao cũng có thể tạm chấp nhận được trên thực tế

Phương pháp tham lam

- Về lý thuyết, có thể chỉ ra những hệ thống tiền tệ mà cách làm này cho ra giải pháp không tối ưu
  - *Hậu quả của phép chọn tham lam* tức thời sẽ tệ hại hơn nếu chúng ta chỉ có 3 loại tiền: 3k, 5k đồng và 8k đồng. Khi đó thuật toán này sẽ thất bại nếu cần đổi **10k** đồng vì khi rút tờ 8k đồng ra rồi thì không có cách nào đổi nốt 2k đồng còn lại nữa

Phương pháp tham lam

- Bài toán trồng cây
  - Một nông dân đang muốn trồng hoa vào khu vườn của mình. Để cho khu vườn trở nên thật màu sắc ông quyết định trồng nhiều loài hoa khác nhau vào khu vườn. Mỗi loài hoa có 1 cách trồng khác nhau do đó ông sẽ trồng từng loài hoa vào các ngày liên tiếp nhau. Cháu của ông rất mong chờ được thấy tất cả loài hoa trong khu vườn đều nở hoa trông sẽ tuyệt vời như thế nào. Tuy nhiên mỗi loài hoa lại có thời gian phát triển từ lúc trồng tới lúc nở hoa khác nhau. Nhiệm vụ của bạn là giúp ông nông dân tìm ra ngày sớm nhất mà tất cả loài hoa đều nở hoa

Phương pháp tham lam

- Bài toán trồng cây
  - Giả sử ông nông dân hiện tại có một số loài hoa

	Loài hoa	Thời gian hoa sẽ nở
1	Hoa hồng	3
2	Hoa lan	4
3	Hoa cúc	2
4	Hoa mười giờ	1

Ý tưởng: Mỗi cách trồng là một hoán vị của *N* (số loài hoa)

Dễ thấy với các loài hoa có thời gian phát triển lâu cần được trồng trước các loài hoa phát triển nhanh (tham lam)
=> Sắp xếp theo thứ tự giảm dần thời gian phát triển của hoa để sinh ra hoán vị tốt nhất

Phương pháp tham lam

- Bài toán trồng cây
  - Giả sử ông nông dân hiện tại có một số loài hoa

	Loài hoa	Thời gian hoa sẽ nở
1	Hoa hồng	3
2	Hoa lan	4
3	Hoa cúc	2
4	Hoa mười giờ	1

Trồng hoa lan (2) vào ngày thứ 1 => sau 1 + 4 = 5 ngày hoa sẽ nở

Trồng hoa hồng (1) vào ngày thứ 2 => sau 2 + 3 = 5 ngày hoa sẽ nở

Trồng hoa cúc (3) vào ngày thứ 3 => sau 3 + 2 = 5 ngày hoa sẽ nở

Trồng mười giờ (4) vào ngày thứ 4 => sau 4 + 1 = 5 ngày hoa sẽ nở

➔ Thời gian sớm nhất tất cả loài hoa đều nở là ngày thứ 6

Phương pháp tham lam

- Bài toán trồng cây
  - Giả sử ông nông dân hiện tại có một số loài hoa

	Loài hoa	Thời gian hoa sẽ nở
1	Hoa hồng	3
2	Hoa lan	4
3	Hoa cúc	2
4	Hoa mười giờ	1

Dễ thấy với các loài hoa có thời gian phát triển lâu cần được trồng trước các loài hoa phát triển nhanh (tham lam) => Sắp xếp theo thứ tự giảm dần thời gian phát triển của hoa để sinh ra hoán vị tốt nhất

Phương pháp tham lam

- Lưu file trên đĩa từ
  - Giả sử bạn có n file trên đĩa từ trong đó file thứ i có dung lượng là L[i]. Gọi π là một hoán vị của $\{1, 2, \ldots, n\}$ tương ứng với một cách lưu trữ file theo thứ tự $\pi(1), \pi(2), \ldots, \pi(n)$. Để truy cập file π(i), bạn phải duyệt qua tất cả các file $\pi(1), \pi(2), \ldots, \pi(i\text{-}1)$. Do đó chi phí để truy cập file π(i) là:

$$C\big(\pi(i)\big) = \sum_{k=1}^{i} L[\pi(k)]$$

Tìm cách lưu trữ file sao cho việc truy xuất được hiệu quả nhất, biết rằng các file có xác suất truy cập như nhau

Phương pháp tham lam

- Lưu file trên đĩa từ
  - Vì các file có xác suất truy cập như nhau, với một cách lưu trữ π cho trước, chi phí để truy xuất ngẫu nhiên một file có kì vọng là:

$$E[cost_{\pi}] = \frac{\sum_{i=1}^{n} C(\pi(i))}{n} = \frac{\sum_{i=1}^{n}\sum_{k=1}^{i} L(\pi(k))}{n}$$

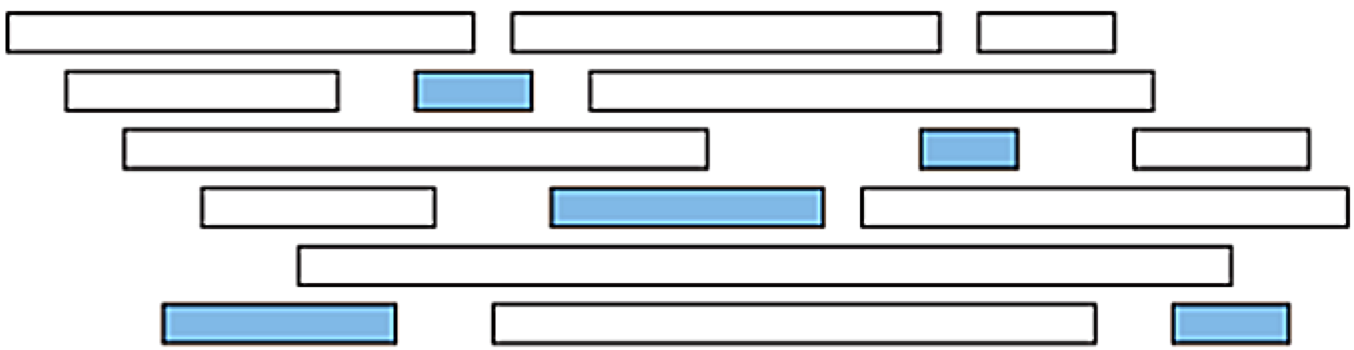
  - Cách lưu trữ file mà chúng ta có thể nghĩ đến ngay đó là cách lưu các file nhỏ ở đầu đĩa từ và lưu các file lớn ở cuối đĩa từ
  - Bằng cách sắp xếp theo chiều tăng của kích thước file, thuật toán có thể được thực thi trong thời gian $O(n\log n)$

Phương pháp tham lam

- Bài toán chọn môn học
  - Trong kì này bạn muốn học *n* môn học, trong đó tiết học của môn học thứ *i* bắt đầu tại thời điểm *S*[*i*] và kết thúc tại thời điểm F[i]. Bạn không được phép chọn hai môn học mà khoảng thời gian của hai môn học gối lên nhau. Làm thế nào để bạn có thể tìm được một tập lớn nhất các môn học không gối lên nhau.

Phương pháp tham lam

- Bài toán chọn môn học
  - Ví dụ: trong hình dưới đây, mỗi môn học được biểu diễn bởi một thanh ngang có chiều dài F[i]−S[i]. Tập các thanh màu xanh chính là một tập có số lượng lớn nhất các môn học không gối lên nhau:



Phương pháp tham lam

- Bài toán chọn môn học
  - Có 3 cách lựa chọn lớp học tham lam mà ta có thể nghĩ đến:
    - Chọn lớp học bắt đầu sớm nhất trước. Tuy nhiên vấn đề của cách chọn này là nếu có một lớp học bắt đầu sớm nhất nhưng lại kéo dài rất lâu thì bạn chỉ có thể chọn được 1 lớp học
    - Chọn lớp học ngắn nhất trước. Vấn đề của cách chọn này là nếu có một lớp học ngắn nhưng gối lên các lớp học bắt đầu trước và sau lớp học này thì sẽ không thể có phương án tối ưu.

Phương pháp tham lam

- Bài toán chọn môn học
  - Có 3 cách lựa chọn lớp học tham lam mà ta có thể nghĩ đến:
    - Chọn lớp học có thời gian hoành thành sớm nhất trước. Có vẻ đây là một cách chọn tham lam tốt vì nếu chọn lớp hoàn thành sớm nhất trước thì sẽ có nhiều thời gian còn lại để chọn các lớp học khác. Thực tế đây là một cách chọn tham lam cho phương án tối ưu.

Phương pháp tham lam

- Bài toán chọn môn học
  - Ví dụ, có 5 môn học với thông tin như sau:

Môn	**Bắt đầu**	**Kết thúc**
1	6	7
2	7	9
3	8	10
4	10	12
5	9	11

  - Phương án tối ưu là {1, 2, 5}

Phương pháp tham lam

- Bài toán lập lịch
  - Bạn được ông chủ giao cho *n* tác vụ và một chiếc máy tính để thực hiện các tác vụ đó. Tác vụ thứ *i* có thời hạn D[i] và nếu bạn hoành thành tác vụ đó trước thời hạn D[i], bạn sẽ được thưởng P[i] đồng, còn nếu hoành thành sau thời hạn D[i] thì bạn sẽ không nhận được đồng tiền thưởng nào cả. Biết rằng mỗi tác vụ mất thời gian đúng 1 đơn vị để hoành thành. Tìm một cách thực thi các tác vụ sao cho bạn nhận được nhiều tiền thưởng nhất. Giả sử thời hạn của các tác vụ là trong khoảng [1, *n*].

Phương pháp tham lam

- Bài toán lập lịch
  - Ý tưởng 30 giây:
    - Bước 1: Xác định tất cả các lịch thực thi từ n tác vụ
    - Bước 2: Tính tổng số tiền thưởng ở mỗi lịch thực thi
    - Bước 3: So sánh số tiền thưởng ở các lịch thực thi và đưa ra lịch cần tìm và tổng số tiền thưởng tương ứng

Phương pháp tham lam

- Bài toán lập lịch
  - Ý tưởng 30 giây:
    - Xếp n công việc vào n thời điểm ta có n! cách nên ta có n! lịch thực thi
    - Ví dụ: với n = 3 → 6 cách

      n = 10 → có 10! = 3 628 800 cách

      n = 60 → có 60! = 3.32×10^{81} cách

      …

      n = 10000 → ?

Phương pháp tham lam

- Bài toán lập lịch
  - Tư tưởng tham lam:
    - Trong tập các tác vụ chưa được chọn, chọn tác vụ có tiền thưởng lớn nhất nếu tác vụ này vẫn còn thực thi được
    - Ta luôn duy trì một danh sách các tác vụ đã chọn
    - Một tác vụ được gọi là thực thi được nếu ta thêm tác vụ này vào danh sách thì vẫn tồn tại một cách thực thi sao cho tất cả các tác vụ được hoàn thành trước thời hạn

Phương pháp tham lam

- Bài toán lập lịch
  - Ví dụ bạn có 5 tác vụ với thông tin như sau:

Tác vụ	Tiền thưởng	Thời hạn
1	100	2
2	19	1
3	27	2
4	25	1
5	15	3

  - Phương án tối ưu là (theo trình tự thực thi) {1,3,5} với tổng tiền thưởng thu được là 142

Phương pháp tham lam

- Bài toán lập lịch
  - Áp dụng với ví dụ trên:
    - Đầu tiên ta chọn tác vụ 1 → thực hiện hết 1 đơn vị thời gian (nhỏ hơn thời hạn là 2)
    - Sau đó là chọn tác vụ 3 với trình tự 1, 3 → hai tác vụ 1 và 3 thực hiện hết 2 đơn vị thời gian
    - Ứng viên tiếp theo là tác vụ 4, tuy nhiên ta không thể thực thi toàn bộ 1, 3, 4 trước thời hạn (hai tác vụ 1 và 3 đã tốn 2 đơn vị thời gian) → bỏ qua 4
    - Ta cũng không chọn Tác vụ 2 được
    - Cuối cùng ta chọn 5

    → Thuật toán tham lam cho ta các tác vụ {1, 3, 5}

Phương pháp tham lam

- Bài toán cái ba lô
  - Cho một cái ba lô có thể đựng một trọng lượng *W* và *n* loại đồ vật, mỗi đồ vật *i* có một trọng lượng *g*[i] và một giá trị *v*[i]. Tất cả các loại đồ vật đều có số lượng không hạn chế. Tìm một cách lựa chọn các đồ vật đựng vào ba lô, chọn các loại đồ vật nào, mỗi loại lấy bao nhiêu sao cho tổng trọng lượng không vượt quá *W* và tổng giá trị là lớn nhất

Phương pháp tham lam

- Bài toán cái ba lô
  - Gọi $X = (X_1, X_2, \ldots., X_n)$ với X_i là số nguyên không âm là một phương án. X_i là số đồ vật thứ i
  - Cần tìm X sao cho:
    - $X_1 \times g[1] + X_2 \times g[2] + \ldots + X_n \times g[n] \leq W$
    - $F(X) = X_1 \times v[1] + X_2 \times v[2] + \ldots + X_n \times v[n]$ --> Max

Phương pháp tham lam

- Bài toán cái ba lô
  - Tư tưởng tham lam theo vật có giá trị lớn nhất
    1. Xét các loại đồ vật theo thứ tự giá trị từ lớn đến nhỏ
    2. Với mỗi đồ vật được xét sẽ lấy một số lượng tối đa mà trọng lượng còn lại của ba lô cho phép
    3. Xác định trọng lượng còn lại của ba lô và quay lại bước 2 cho đến khi không còn có thể chọn được đồ vật nào nữa

Phương pháp tham lam

- Bài toán cái ba lô
  - Ví dụ: Một ba lô có thể đựng một trọng lượng là 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng được cho trong bảng bên dưới:

Loại đồ vật	**Trọng lượng**	**Giá trị**
A	15	30
B	10	25
C	2	2
D	4	6

Phương pháp tham lam

- Bài toán cái ba lô
  - Ví dụ:

ĐV	**TL**	**GT**
a	15	30
b	10	25
d	4	6
c	2	2

- Phương án là X=(Xa,Xb,Xc,Xd)
- Xa = 37/15 = 2
- *W* = 37 - 2*15 = 7
- Xb = 7/10 =0
- Xd = 7/4 = 1
- *W* = 7-4 = 3
- Xc = 3/2 = 1
- *W* = 3 – 2 = 1
- TTL là 2*15 + 1*4 + 1*2 = 36
- TGT là 2*30+1*6+1*2 = 68

Phương pháp tham lam

- Bài toán cái ba lô
  - Tư tưởng tham lam theo đơn giá
    1. Tính đơn giá (giá cho một đơn vị trọng lượng) cho các loại đồ vật
    2. Xét các loại đồ vật theo thứ tự đơn giá từ lớn đến nhỏ
    3. Với mỗi đồ vật được xét sẽ lấy một số lượng tối đa mà trọng lượng còn lại của ba lô cho phép
    4. Xác định trọng lượng còn lại của ba lô và quay lại bước 3 cho đến khi không còn có thể chọn được đồ vật nào nữa

Phương pháp tham lam

- Bài toán cái ba lô
  - Ví dụ: Một ba lô có thể đựng một trọng lượng là 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng được cho trong bảng bên dưới:

Loại đồ vật	**Trọng lượng**	**Giá trị**
A	15	30
B	10	25
C	2	2
D	4	6

Phương pháp tham lam

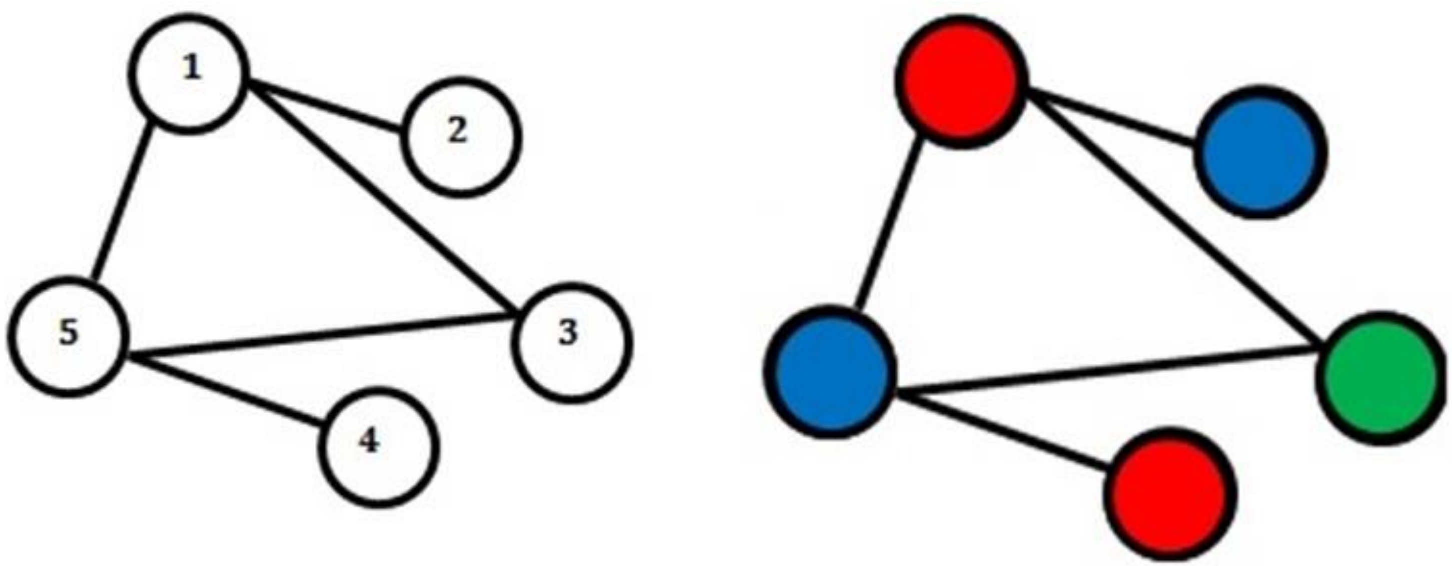
- Bài toán cái ba lô
  - Ví dụ:

ĐV	TL	GT	ĐG
b	10	25	2.5
a	15	30	2.0
d	4	6	1.5
c	2	2	1.0

- Phương án là X=(Xa,Xb,Xc,Xd)
- Xb = 37/10 = 3
- *W*= 37 - 3*10 = 7
- Xa = 7/15 =0
- Xd = 7/4 = 1
- *W* = 7-4 = 3
- Xc = 3/2 = 1
- *W* = 3 – 2 = 1
- TTL là 3*10 + 1*4 + 1*2 = 36
- TGT là 3*25+1*6+1*2 = 83

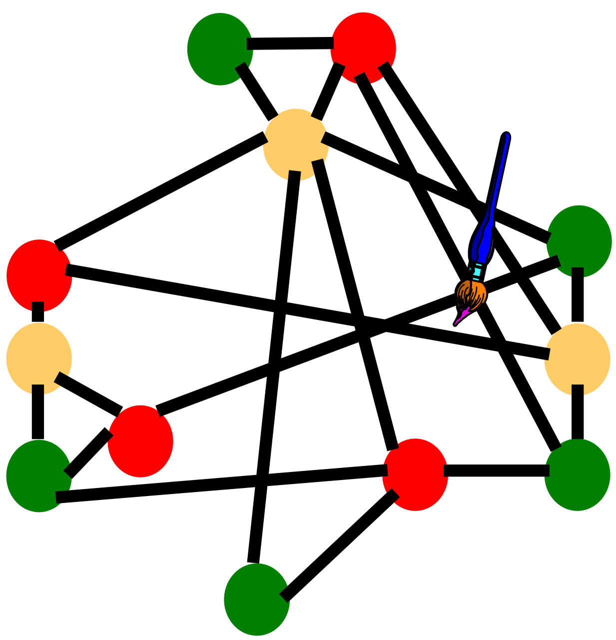
Tô màu đồ thị

- Bài toán tô màu đồ thị
  - Tô màu (đỉnh) đồ thị là việc thực hiện gán màu cho mỗi đỉnh của đồ thị, sao cho hai đỉnh kề nhau không cùng một màu và số màu được sử dụng là ít nhất
  - **Số màu ít nhất** có thể sử dụng để tô màu đồ thị được gọi là **sắc số** (chromatic number) của đồ thị đó



Tô màu đồ thị

- Thuật toán tham lam cơ bản tô màu đồ thị
  - Bài toán tô màu đồ thị với số màu tối thiểu thuộc lớp bài toán **NP-đầy đủ**
  - **Có thể giải bằng cách duyệt hết mọi khả năng! (chỉ áp dụng với các đồ thị nhỏ)**



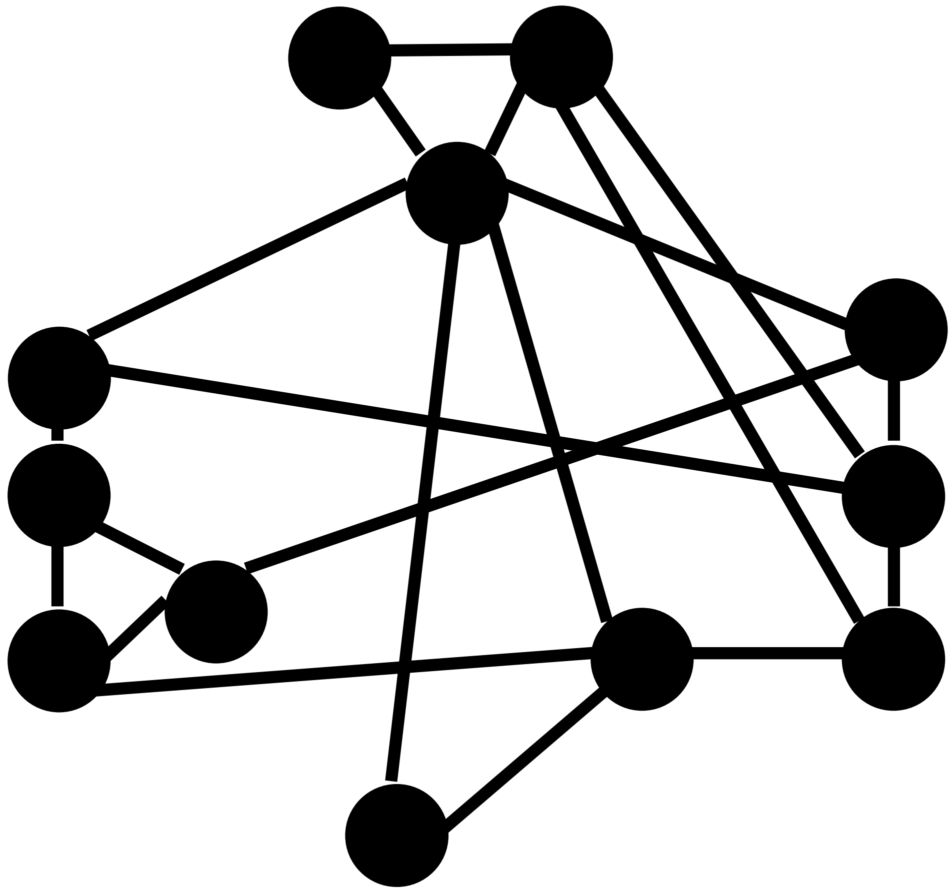
Tô màu đồ thị

- Thuật toán tham lam cơ bản tô màu đồ thị
  - Thuật toán tham lam cơ bản không sử dụng nhiều hơn *d* + 1 màu với *d* là bậc lớn nhất của đồ thị
  - **Thuật toán**

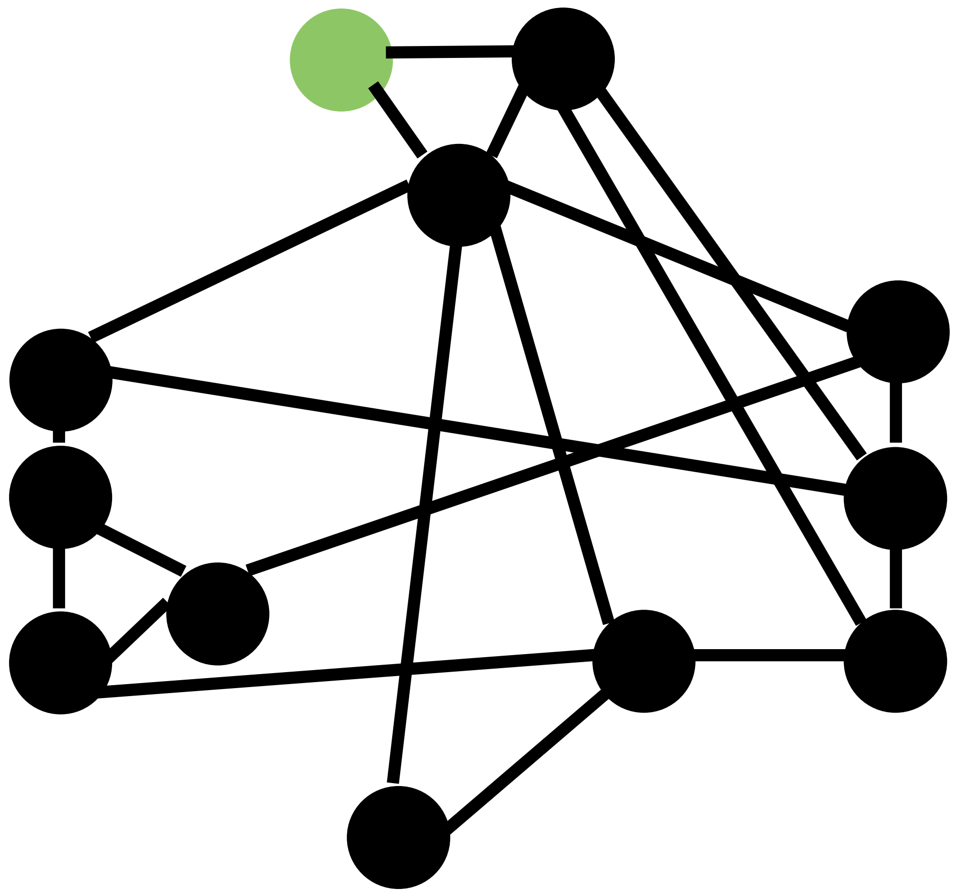
B1. Tô màu đỉnh đầu tiên với màu thứ nhất.

B2. Thực hiện các thao tác sau với V - 1 đỉnh còn lại. Xem xét đỉnh được chọn hiện thời *v* và tô màu đỉnh *v* với màu được đánh số thấp nhất mà chưa được sử dụng để tô bất kỳ đỉnh nào liền kề với *v* trước đó. Nếu tất cả các màu đã được sử dụng trước đó xuất hiện trên đỉnh liền kề với *v*, gán một màu mới cho nó.

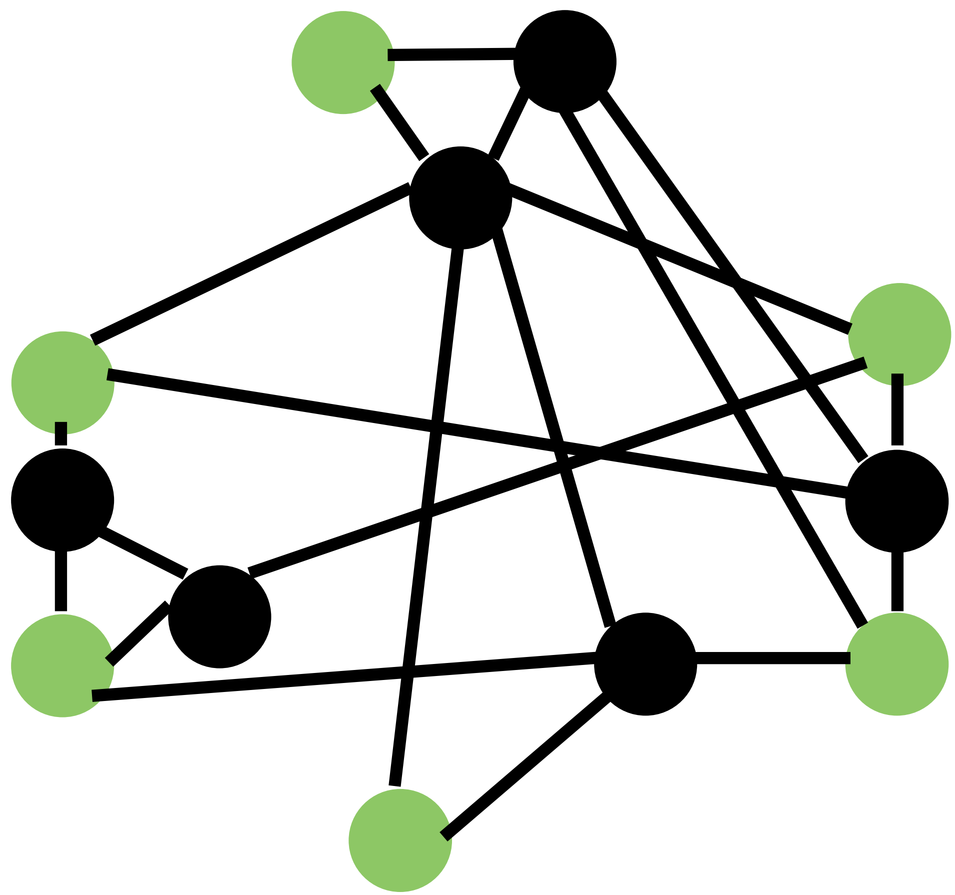
Tô màu đồ thị



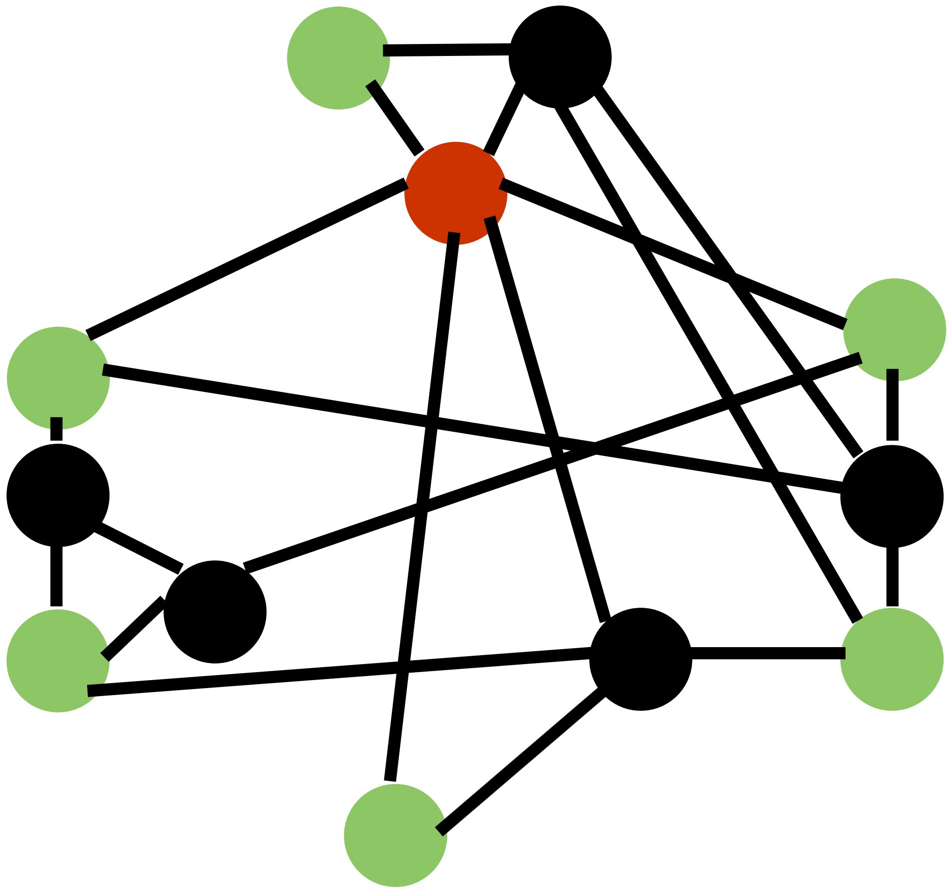
Tô màu đồ thị



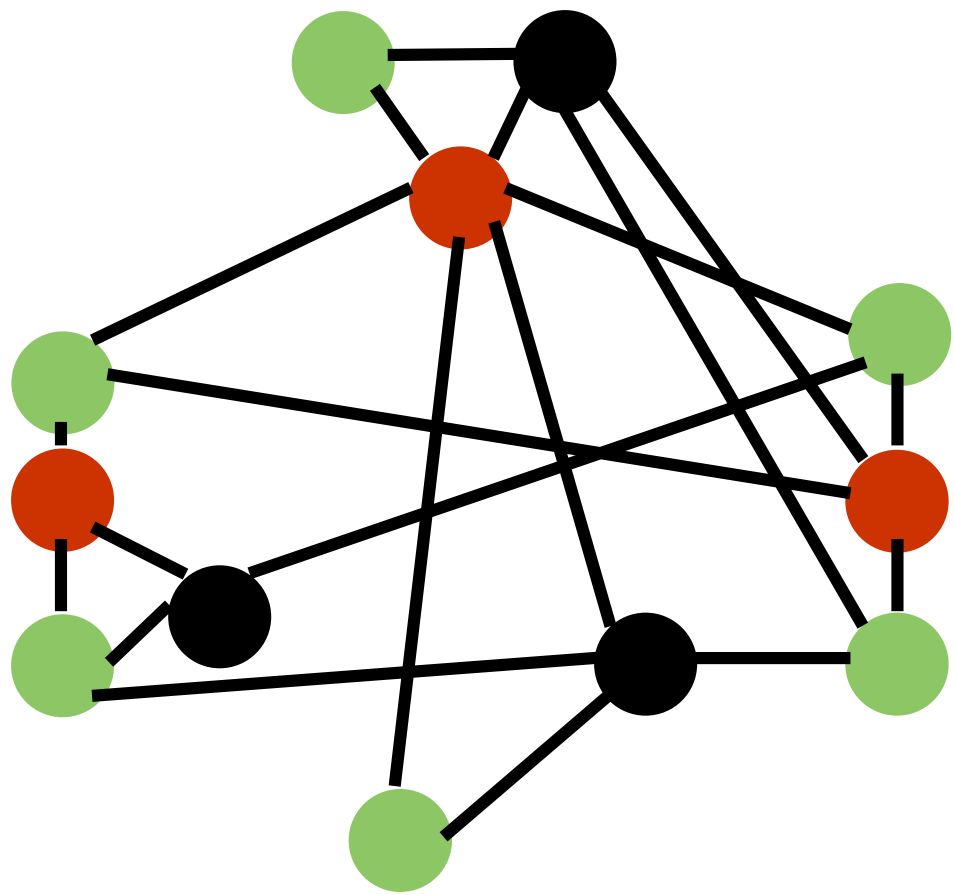
Tô màu đồ thị



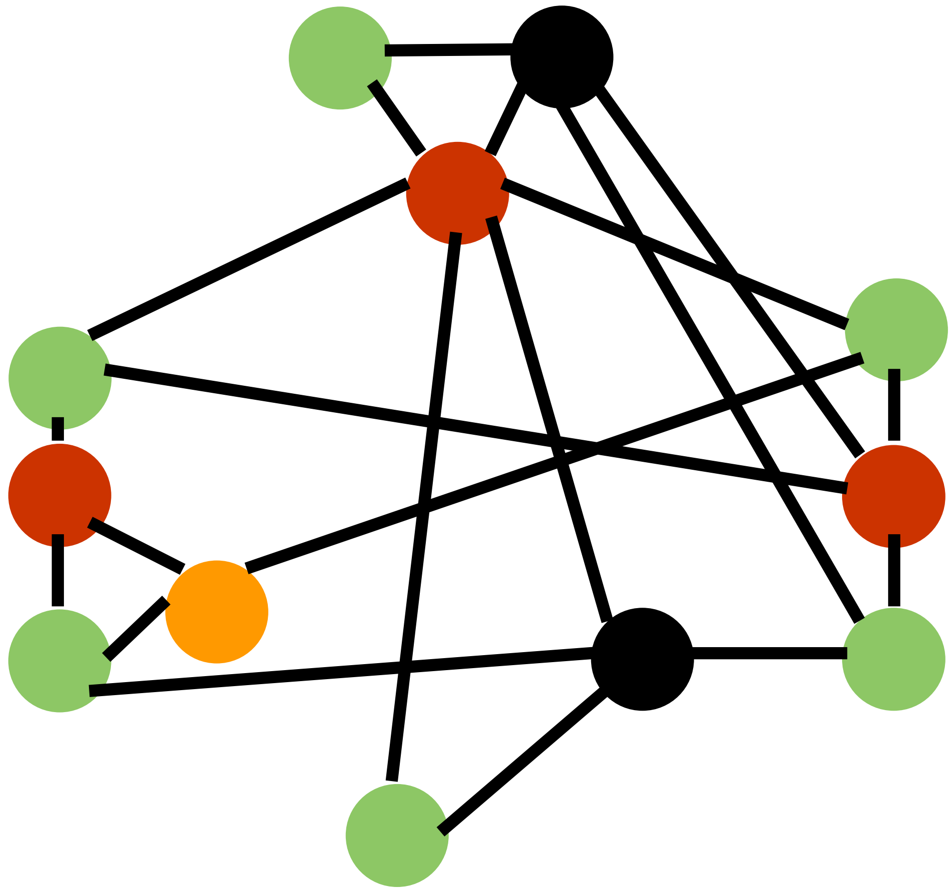
Tô màu đồ thị



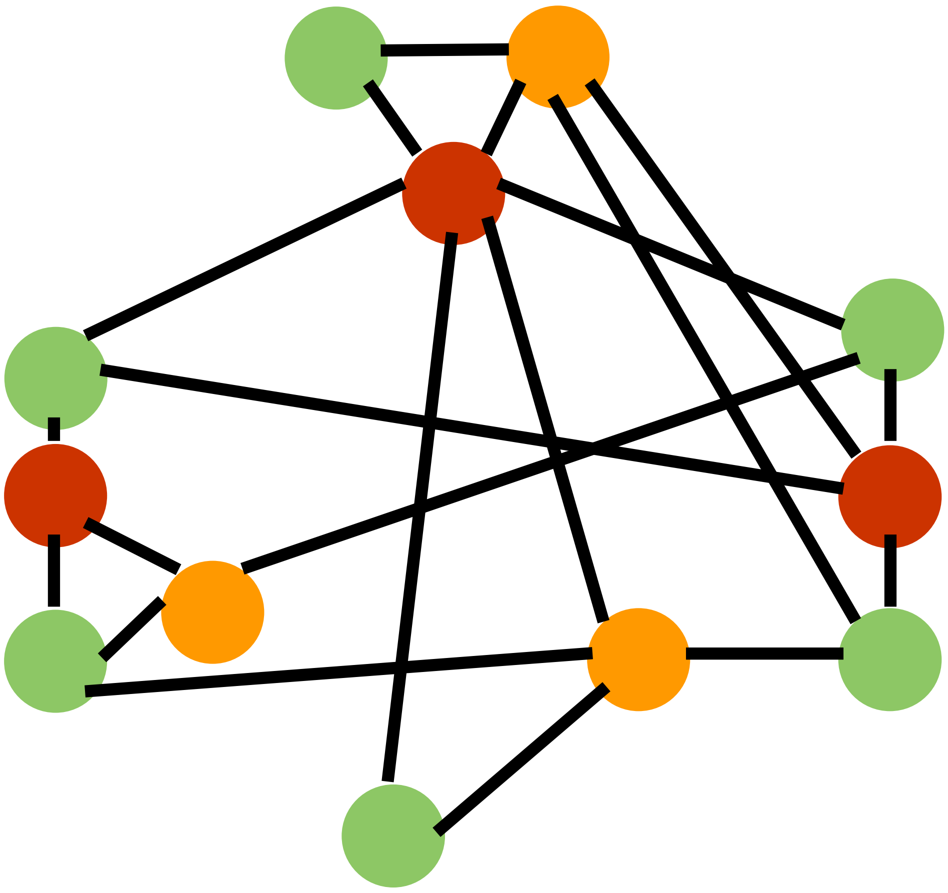
Tô màu đồ thị



Tô màu đồ thị

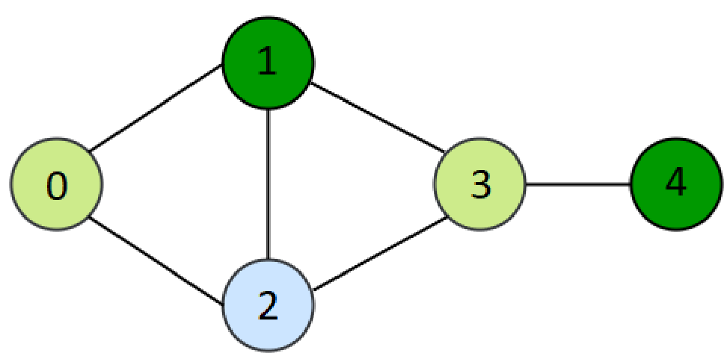


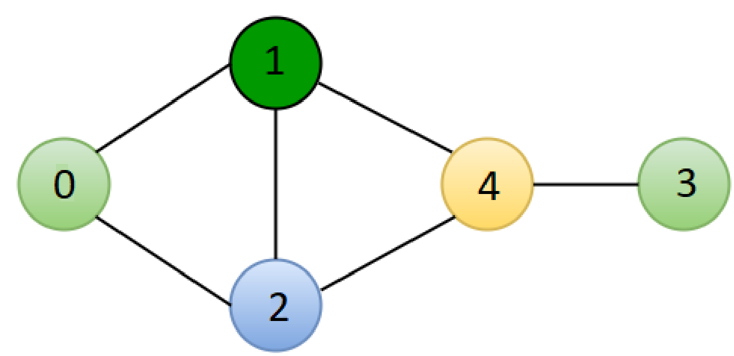
Tô màu đồ thị



Tô màu đồ thị

- Thuật toán tham lam cơ bản tô màu đồ thị
  - **Phân tích thuật toán**
  - Độ phức tạp về thời gian: $O(V^2 + E)$
  - Không phải lúc nào cũng sử dụng số màu ít nhất
  - Số màu được sử dụng đôi khi phụ thuộc vào **thứ tự các đỉnh được tô** → thứ tự các đỉnh được tô là quan trọng → cần tìm cách sắp xếp các đỉnh được tô để thuật toán làm việc hiệu quả hơn





Tô màu đồ thị

- Thuật toán Welsh Powell
  - **Ý tưởng**: bắt đầu với các đỉnh có bậc lớn nhất → xử lý các đỉnh có khả năng **xung đột** sớm nhất có thể

Bước 1. Tính giá trị bậc của các đỉnh trong V. Lập danh sách V' = [v_1, v_2, ...,v_n] là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần: $d(v_1) > d(v_2) > ... > d(v_n)$. Gán màu $c = 1$

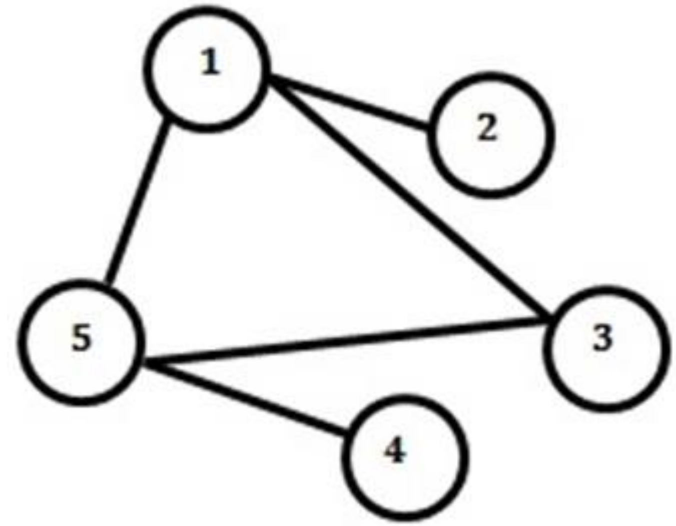
Bước 2. Tô màu *c* cho đỉnh đầu tiên trong danh sách V'. Duyệt lần lượt các đỉnh khác trong V' và chỉ tô màu *c* cho các đỉnh không kề đỉnh đã có màu *c*.

Bước 3. Kiểm tra nếu tất cả các đỉnh trong V đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng *c* màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang bước 4.

Bước 4. Loại khỏi danh sách V' các đỉnh đã được tô màu. Các đỉnh trong V' vẫn theo thứ tự bậc giảm dần. Gán $c = c + 1$ và quay lại bước 2

Tô màu đồ thị

- Thuật toán Welsh Powell
  - **Ví dụ**



Bước 1. Sau sắp xếp V' có thứ tự là [1, 5, 3, 2, 4]. Gán c = 1

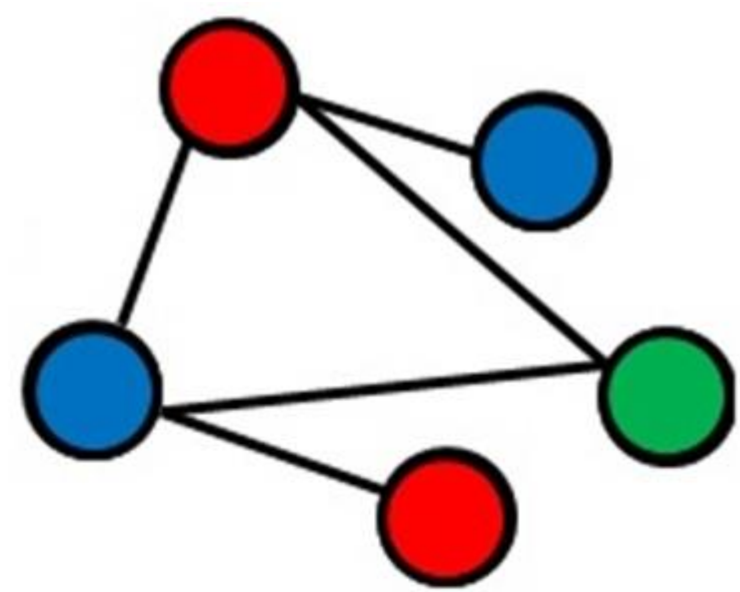
Bước 2. Tô màu 1 (đỏ) cho đỉnh 1. Tô các đỉnh còn lại trong V'. Đỉnh 2, 3 và 5 kề đỉnh 1 nên chưa tô màu cho các đỉnh này. Đỉnh 4 không kề với đỉnh 1 → thực hiện tô màu 1 (đỏ) cho đỉnh 4.

Bước 3. Kiểm tra thấy vẫn còn các đỉnh trong V chưa được tô màu nên chuyển sang bước 4

Bước 4. Loại bỏ các đỉnh 1, 4 đã được tô màu ra khỏi V' ta thu được V' = [5, 3, 2]. Đổi màu mới → c = 2 và lặp lại từ bước 2

Tô màu đồ thị

- Thuật toán Welsh Powell
  - **Ví dụ**
  - Cuối cùng ta thu được đồ thị đã được tô bởi 3 màu

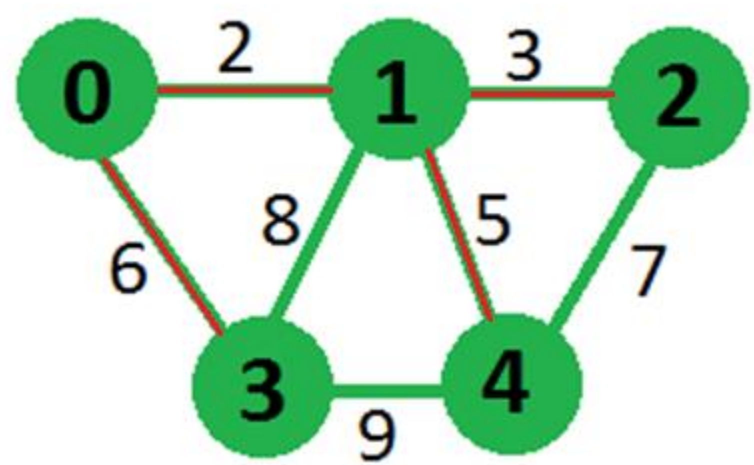


Cây bao trùm

- Cho một đồ thị vô hướng hoặc có hướng, một cây bao trùm (cây khung) của đồ thị đã cho là một đồ thị con và là một **cây** kết nối tất cả các đỉnh của đồ thị
- Một đơn đồ thị có thể có nhiều cây bao trùm
- Cây bao trùm tối thiểu (MST) của một đồ thị vô hướng có trọng số và liên thông là cây bao trùm có tổng trọng số nhỏ hơn hoặc bằng tổng trọng số của bất kỳ cây bao trùm nào khác
- Một cây bao trùm tối thiểu có (V - 1) cạnh, trong đó V là số đỉnh của đồ thị

Cây bao trùm

- Cây bao trùm tối thiểu (MST)



- Cây có tổng trọng số nhỏ nhất là 16 với các cạnh là:
  0-1, 1-2, 0-3 và 1-4

Cây bao trùm

- Thuật toán Kruskal

  Bước 1. Sắp xếp tất cả các cạnh của đồ thị theo thứ tự tăng dần của các trọng số.

  Bước 2. Chọn cạnh có trọng số nhỏ nhất. Kiểm tra xem nó có **tạo thành một chu trình** đối với cây bao trùm hiện đang xem xét hay không. Nếu không thì đưa cạnh đó vào cây bao trùm, ngược lại thì loại bỏ cạnh này.

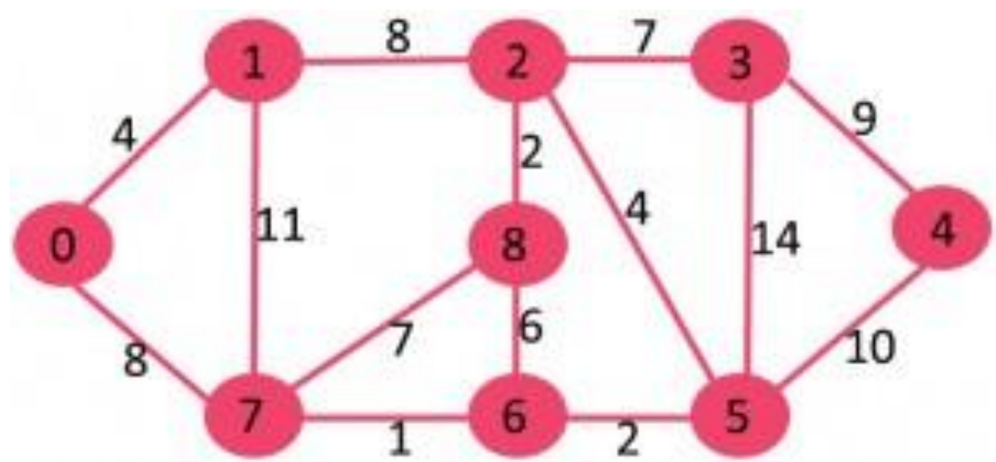
  Bước 3. Lặp lại Bước 2 chi tới khi có V-1 cạnh trong cây bao trùm.

Độ phức tạp về thời gian là O(ElogE) hoặc O(ElogV)

Trong **bước 2** của thuật toán cần áp dụng thuật toán phát hiện **chu trình** trong đồ thị

Cây bao trùm

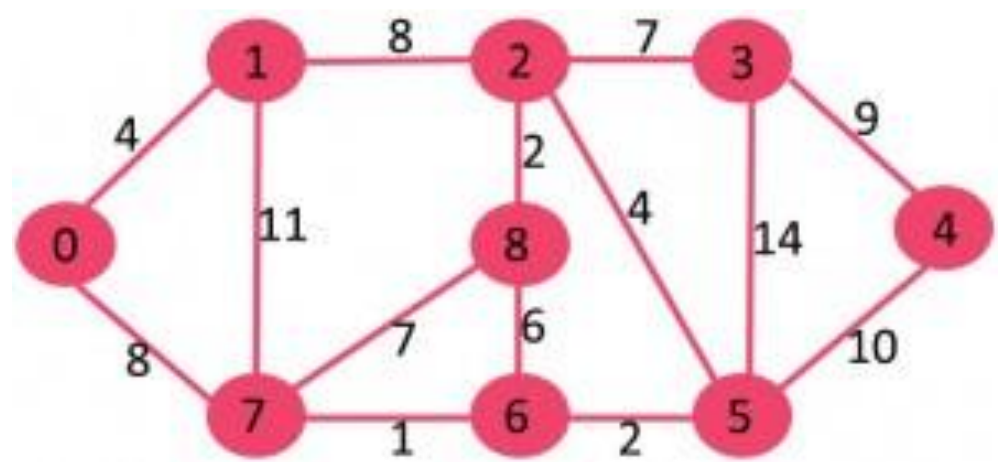
- Thuật toán Kruskal
  - Là thuật toán tham lam → chọn cạnh có trọng số nhỏ nhất không tạo thành chu trình để xây dựng cây
  - Ví dụ, ta có đồ thị như hình dưới



Cây bao trùm

- Thuật toán Kruskal
  - Sau khi sắp xếp lại các trọng số ta có:

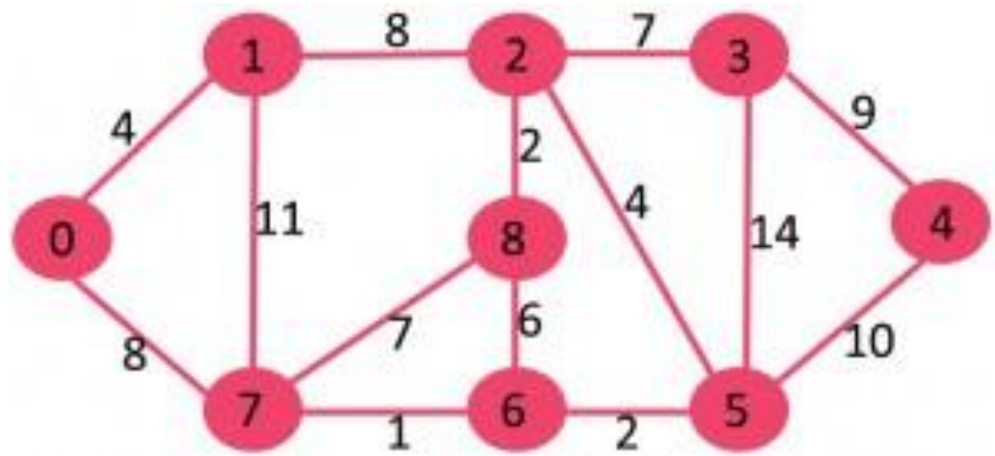
Weight	Src	Dest
1	7	6
2	8	2
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7
8	1	2
9	3	4
10	5	4
11	1	7
14	3	5



Cây bao trùm

- Thuật toán Kruskal

Thực hiện chọn lần lượt các cạnh từ danh sách đã được sắp xếp

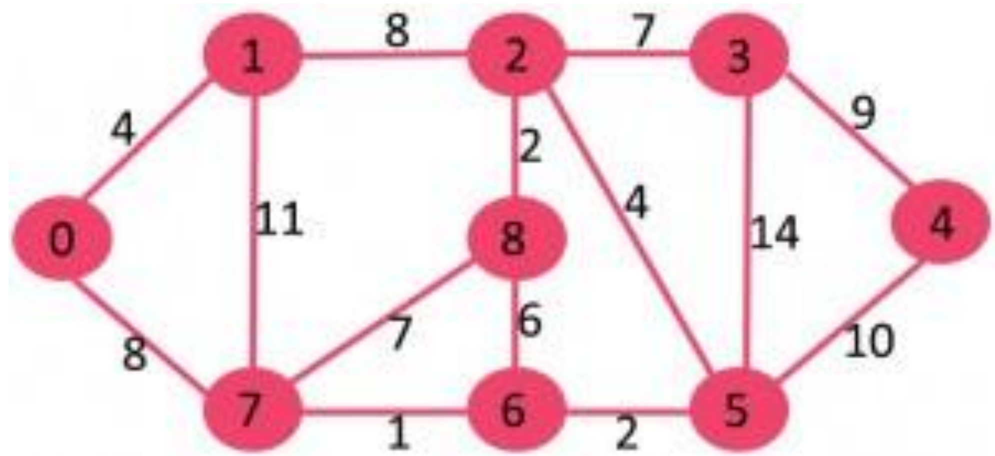


1. *Chọn cạnh 7-6:* Không tạo thành chu trình, đưa vào MST.

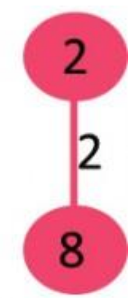
Cây bao trùm

- Thuật toán Kruskal

Thực hiện chọn lần lượt các cạnh từ danh sách đã được sắp xếp



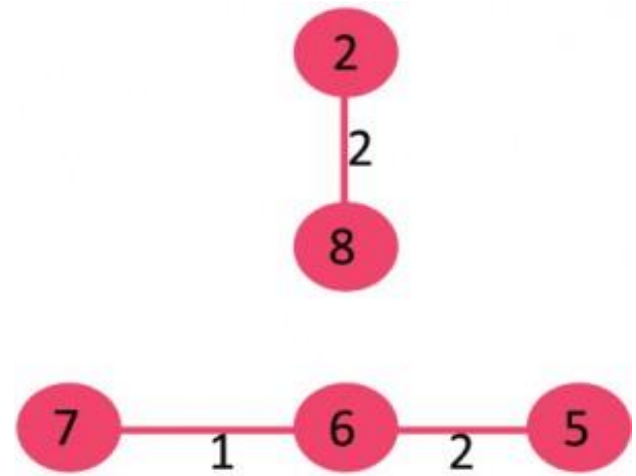
2. *Chọn cạnh 8-2:* Không tạo thành chu trình, đưa vào MST.



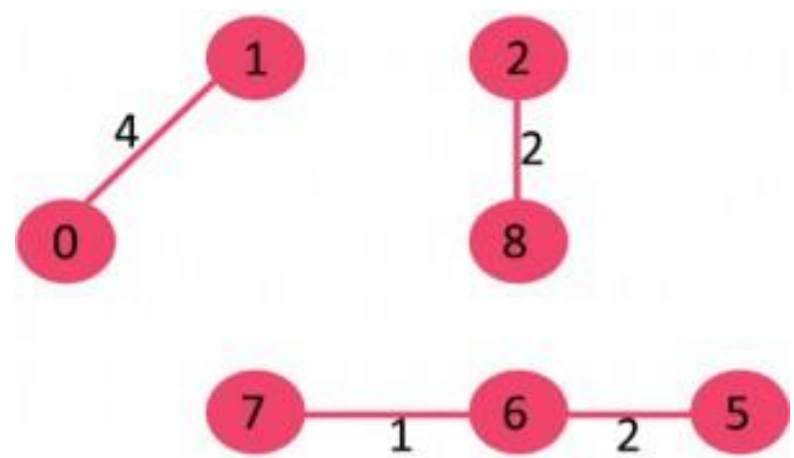
Cây bao trùm

- Thuật toán Kruskal

3. *Chọn cạnh 6-5:* Không tạo thành chu trình, đưa vào MST.

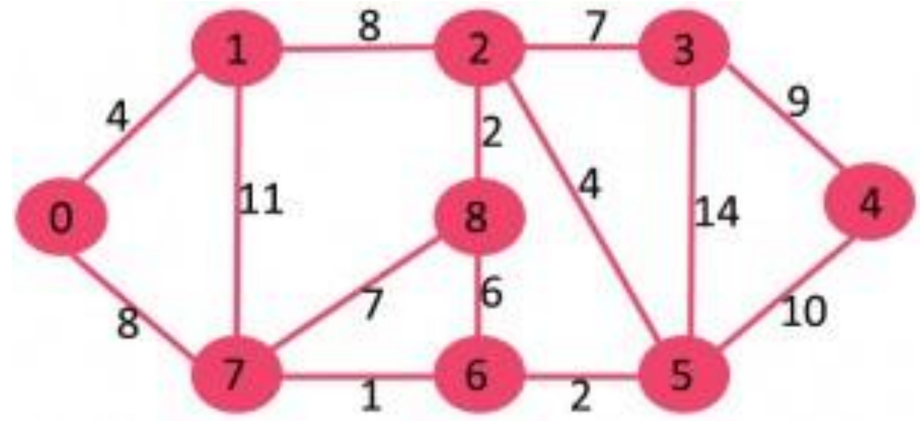


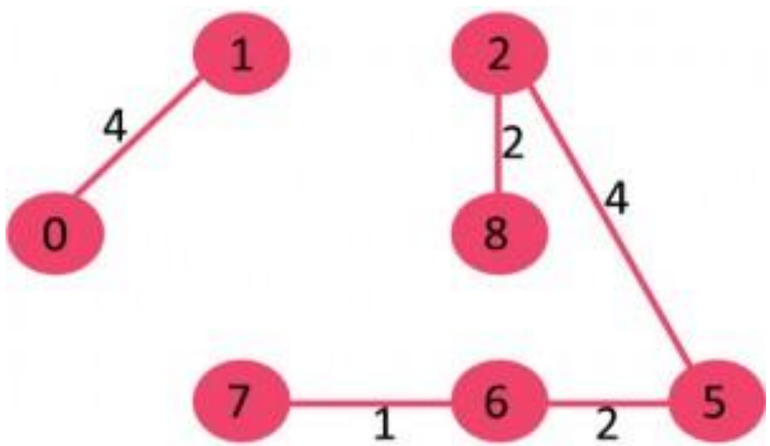
4. *Chọn cạnh 0-1:* Không tạo thành chu trình, đưa vào MST.



Cây bao trùm

- Thuật toán Kruskal



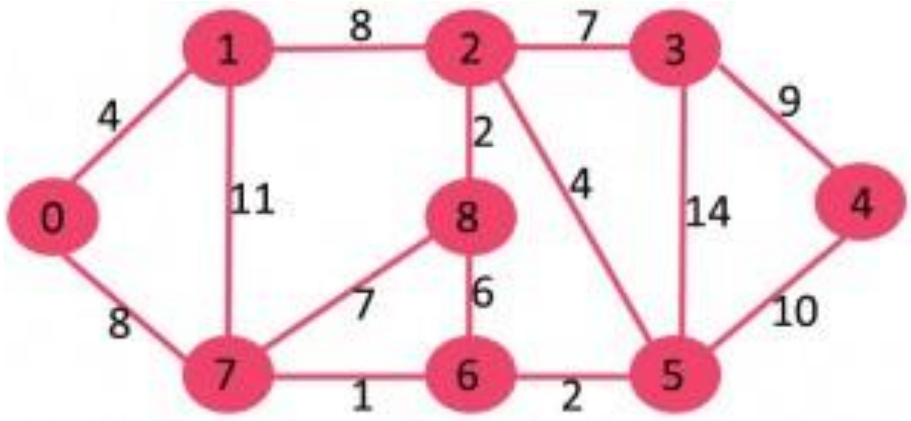


5. *Chọn cạnh 2-5:* Không tạo thành chu trình, đưa vào MST.

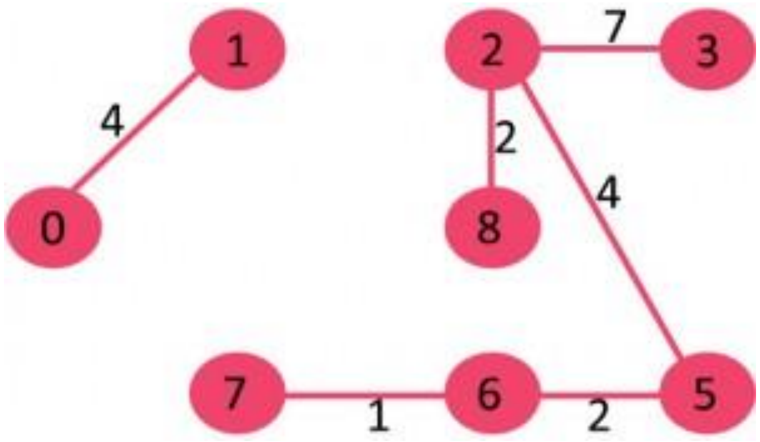
6. *Chọn cạnh 8-6:* Tạo thành chu trình → loại bỏ.

Cây bao trùm

- Thuật toán Kruskal



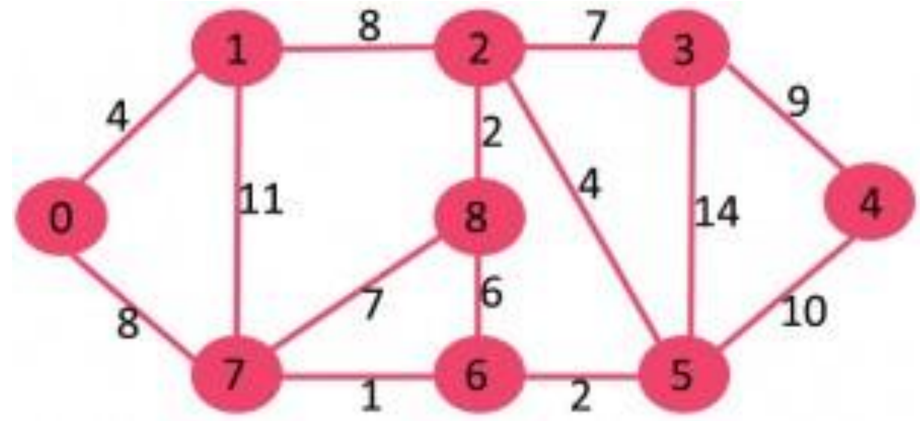
7. *Chọn cạnh 2-3:* Không tạo thành chu trình, đưa vào MST.



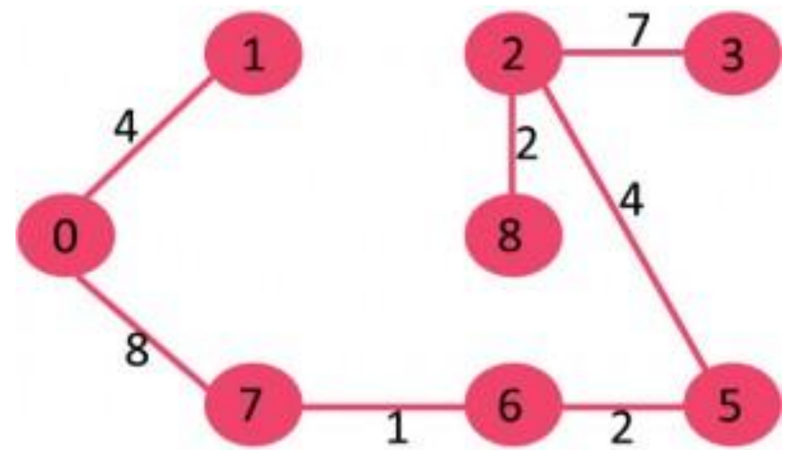
8. *Chọn cạnh 7-8:* Tạo thành chu trình → loại bỏ

Cây bao trùm

- Thuật toán Kruskal



9. *Chọn cạnh 0-7:* Không tạo thành chu trình, đưa vào MST.

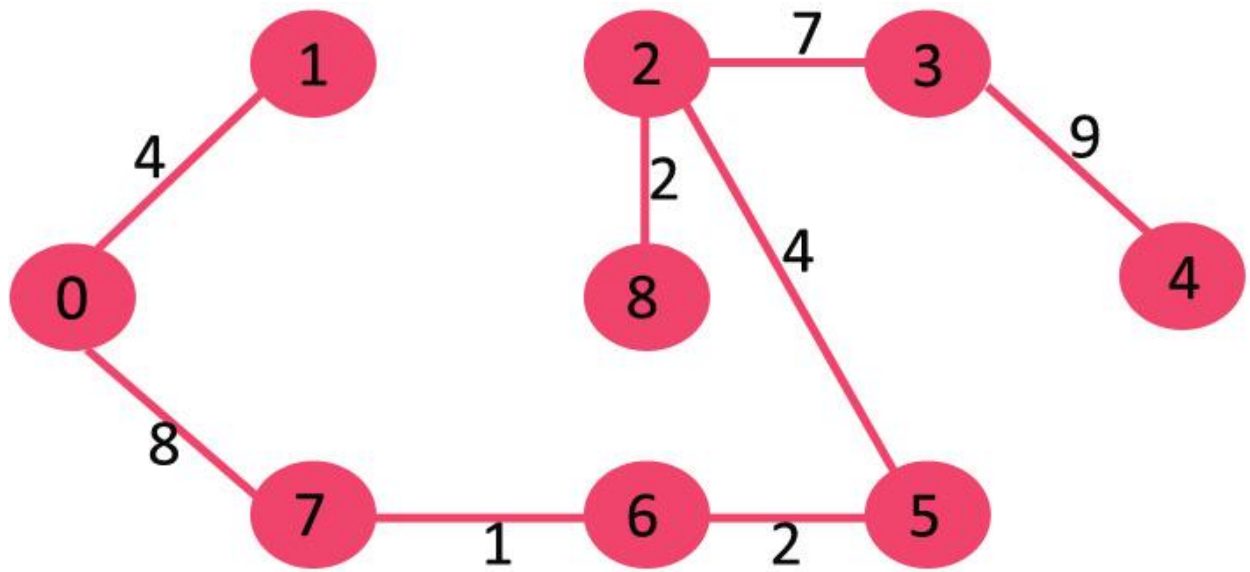


10. *Chọn cạnh 1-2:* Tạo thành chu trình → loại bỏ

Cây bao trùm

- Thuật toán Kruskal

11. *Chọn cạnh 3-4:* Không tạo thành chu trình, đưa vào MST.



Đã chọn được |V|-1 = 7 cạnh → dừng thuật toán

Cây bao trùm

- Thuật toán Prim
  - Là thuật toán theo chiến lược tham lam
  - Bắt đầu với MST rỗng
  - Duy trì hai tập đỉnh, tập thứ nhất là các đỉnh **nằm trong MST** và tập thứ hai **chưa** được đưa vào MST
  - Mỗi bước xem xét **các cạnh nối hai tập đỉnh** và chọn đỉnh có trọng số **nhỏ nhất**
  - Đưa cạnh được chọn vào MST
  - Độ phức tạp về thời gian là $O(|V|^2)$

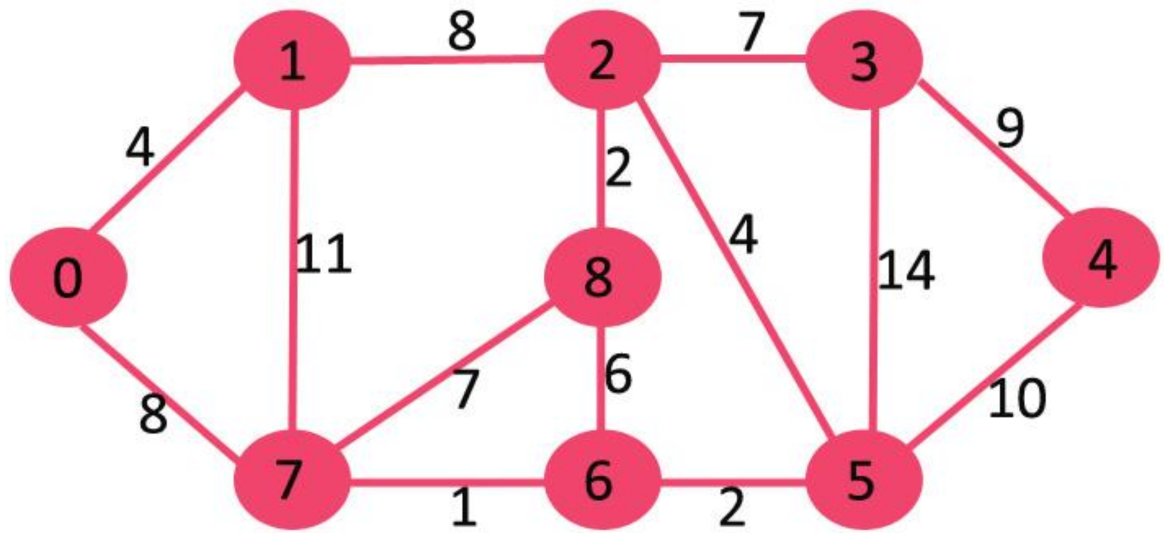
Cây bao trùm

- Thuật toán Prim

Bước 1. Khởi tạo cấu trúc mstSet lưu các đỉnh đã nằm trong MST
Bước 2. Gán khóa cho các đỉnh giá trị **dương vô cùng**, đỉnh được chọn đầu tiên có khóa là 0
Bước 3. Lặp cho đến khi mstSet chứa tất cả các đỉnh
Bước 3.1. Chọn đỉnh u chưa nằm trong mstSet và có giá trị khóa nhỏ nhất
Bước 3.2. Thêm u vào mstSet
Bước 3.3. Cập nhật giá trị khóa của tất cả các đỉnh kề u. Với mỗi đỉnh v kề u, nếu trọng số của cạnh u-v nhỏ hơn giá trị khóa trước đó của v thì cập nhật giá trị khóa là trọng số của u-v

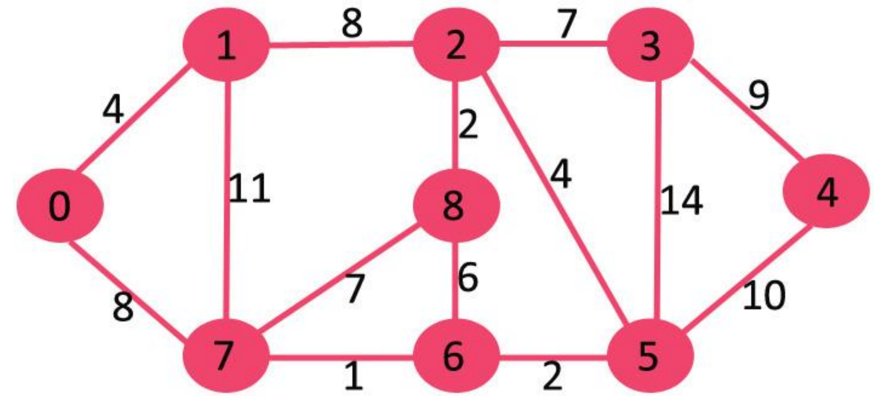
Cây bao trùm

- Thuật toán Prim
  - **Ví dụ**:



  - Thiết lập mstSet = {0, INF, INF, INF, INF, INF, INF, INF} với INF là giá trị dương vô cùng

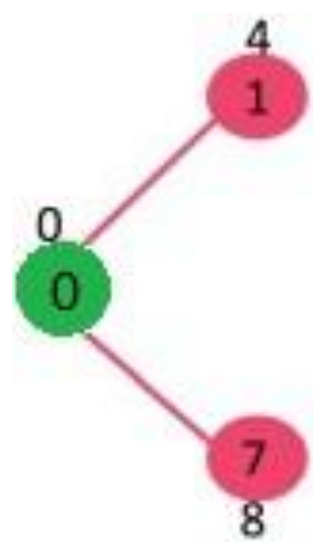
Cây bao trùm



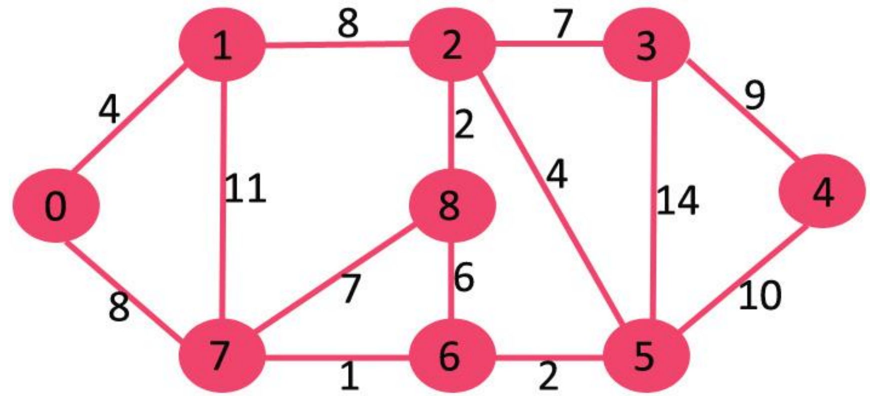
- Thuật toán Prim
  - **Ví dụ**:

Chọn đỉnh 0 (có khóa nhỏ nhất), thêm vào mstSet.
→ mstSet = {0}

Cập nhật khóa của tất cả các đỉnh kề 0 là 1 và 7
→ Khóa của các đỉnh 1 và 7 thành 4 và 8



Cây bao trùm

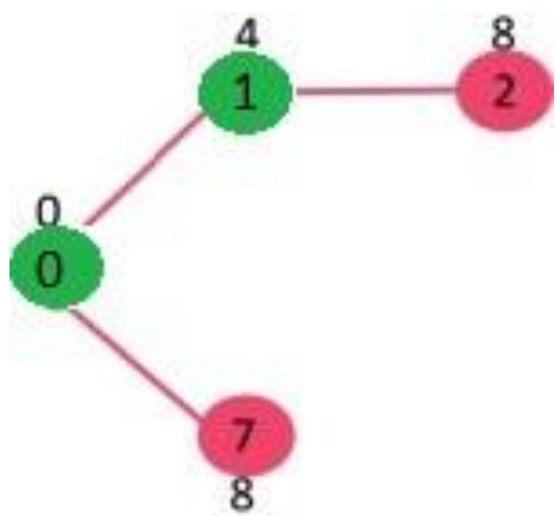


- Thuật toán Prim
  - **Ví dụ**:

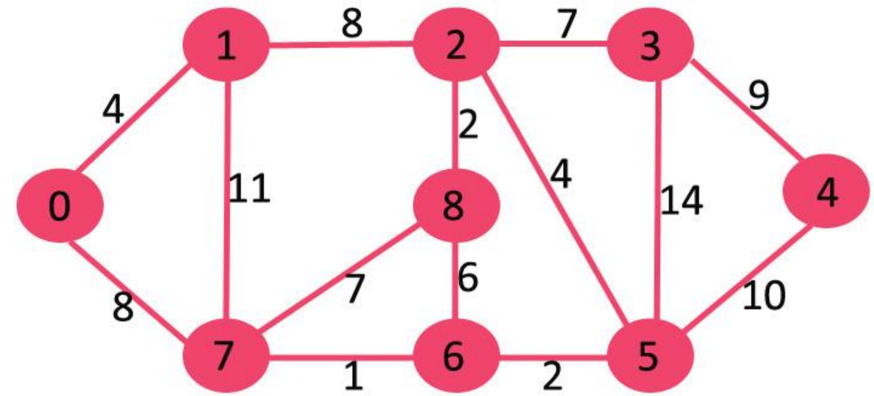
Chọn đỉnh có khóa nhỏ nhất không nằm trong MST.
→ Đỉnh 1 được chọn và thêm vào mstSet

→ mstSet = {0, 1}

Cập nhật khóa của tất cả các đỉnh kề 1 là **8**
→ Khóa của đỉnh 2 thành 8



Cây bao trùm

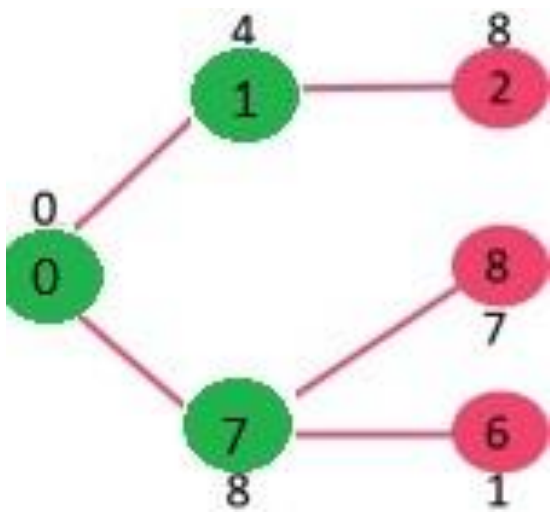


- Thuật toán Prim
  - **Ví dụ**:

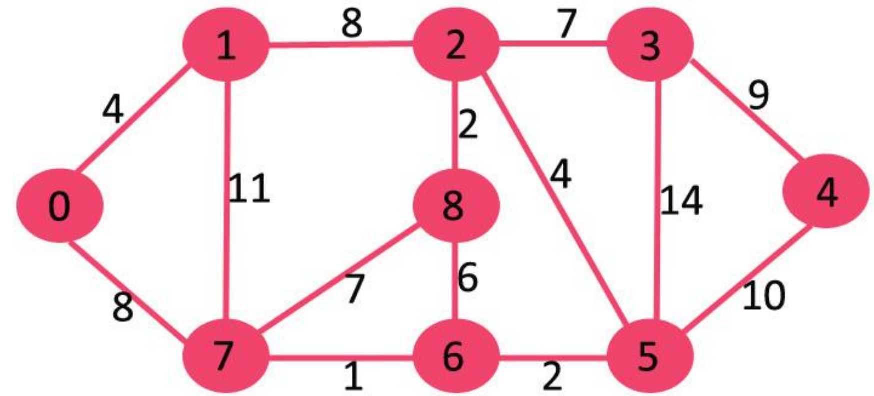
Chọn đỉnh có khóa nhỏ nhất không nằm trong MST.
→ Chọn đỉnh 2 hoặc 7, chọn 7 và thêm vào mstSet

→ mstSet = {0, 1, 7}

Cập nhật khóa của tất cả các đỉnh kề 7 là **6** và **8**
→ Khóa của các đỉnh 6 và 8 thành 1 và 7



Cây bao trùm

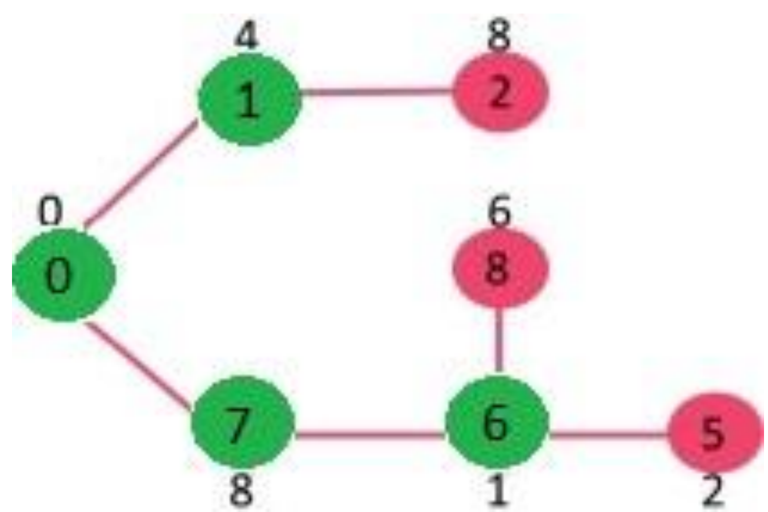


- Thuật toán Prim
  - **Ví dụ**:

Chọn đỉnh có khóa nhỏ nhất không nằm trong MST.
→ Chọn đỉnh 6 và thêm vào mstSet

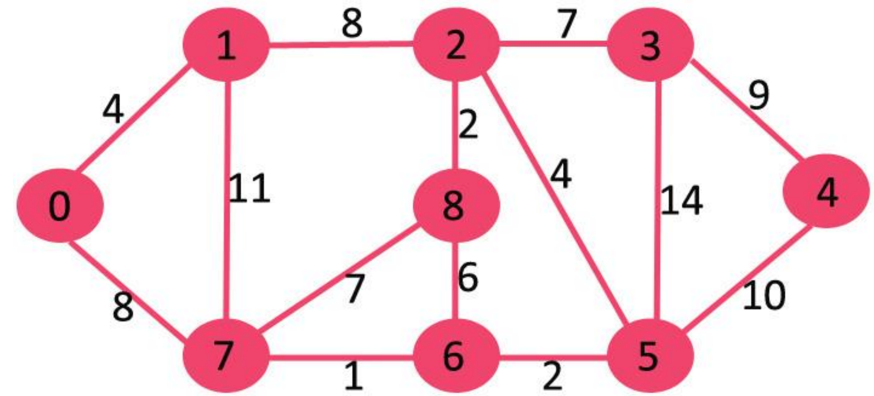
→ mstSet = {0, 1, 7, 6}

Cập nhật khóa của tất cả các đỉnh kề 6 là **5** và **8**
→ Khóa của các đỉnh 5 và 8 thành 2 và 6



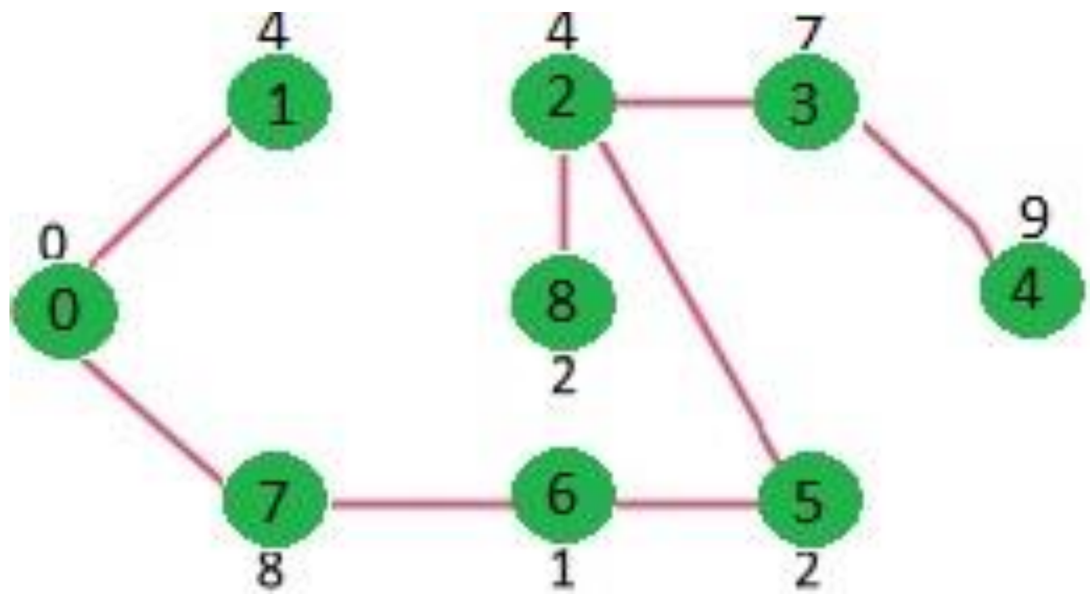
Cây bao trùm

- Thuật toán Prim
  - **Ví dụ**:



Tiếp tục các bước chọn cạnh như vậy cho đến khi tất cả các đỉnh của đồ thị nằm trong mstSet

Cuối cùng ta thu được cây bao trùm nhỏ nhất



TỔNG KẾT

1. Tìm hiểu về thuật toán tham lam
2. Một số bài toán minh họa
3. Ưu điểm:
   - Dễ tiếp cận, đặc biệt là các bài toán đồ thị và NP-đầy đủ
   - Thời gian chạy → phù hợp với bài toàn cần giải quyết
   - Nếu chứng minh được lời giải tham lam là tối ưu toàn cục cho một lớp bài toán thì sẽ ưu tiên được chọn
4. Nhược điểm:
   - Rất khó chứng minh một lời giải tham lam là tối ưu
   - Đa số các lời giải tham lam là không tối ưu

Bài tập

- Máy rút tiền ATM
  - Giả sử có các loại tiền: 100.000 vnđ, 50.000 vnđ, 40.000 vnđ và 10.000 vnđ
  - Mỗi loại tiền đều có số lượng không hạn chế
  - Khách hàng cần rút một số tiền ***n*** vnđ (tính chẵn đến 10.000 vnđ, hay ***n*** chia hết cho 10.000)
  - Phương án trả tiền sao cho trả đủ ***n*** vnđ và số tờ tiền phải trả là ít nhất

Bài tập

- Sắp xếp công việc
  - Cho n công việc phải hoàn thành, mỗi công việc thực hiện trong một đơn vị thời gian. Việc i đem lại w_i tiền thưởng nếu hoàn thành đúng hạn d_i.
  - Tìm cách thực hiện các công việc để có lợi nhuận cao nhất mà thời gian thực hiện là ít nhất

Bài tập

- Bài toán đóng thùng
  - Input:
    - N số nguyên dương
    - B Kích thước thùng
    - $A_1, \ldots, A_N$ kích thước N vật $A_i \leq B$.
  - Output
    - Số thùng ít nhất chứa tất cả các vật