



## Bài 2

# BIỂU DIỄN DỮ LIỆU ĐỊA LÝ

Ths Trần Mạnh Trường  
Truongtm@gmail.com

Hà Nội – 2020





# Các chủ đề nghiên cứu

1

Giới thiệu về GIS

2

**Biểu diễn dữ liệu địa lý**

3

Định vị và tham chiếu không gian

4

Thu thập dữ liệu không gian

5

Phân tích dữ liệu không gian

6

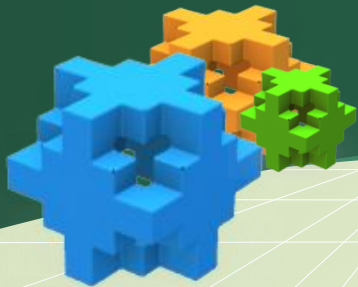
Hiển thị dữ liệu địa lý

7

Các chủ đề GIS nâng cao

8

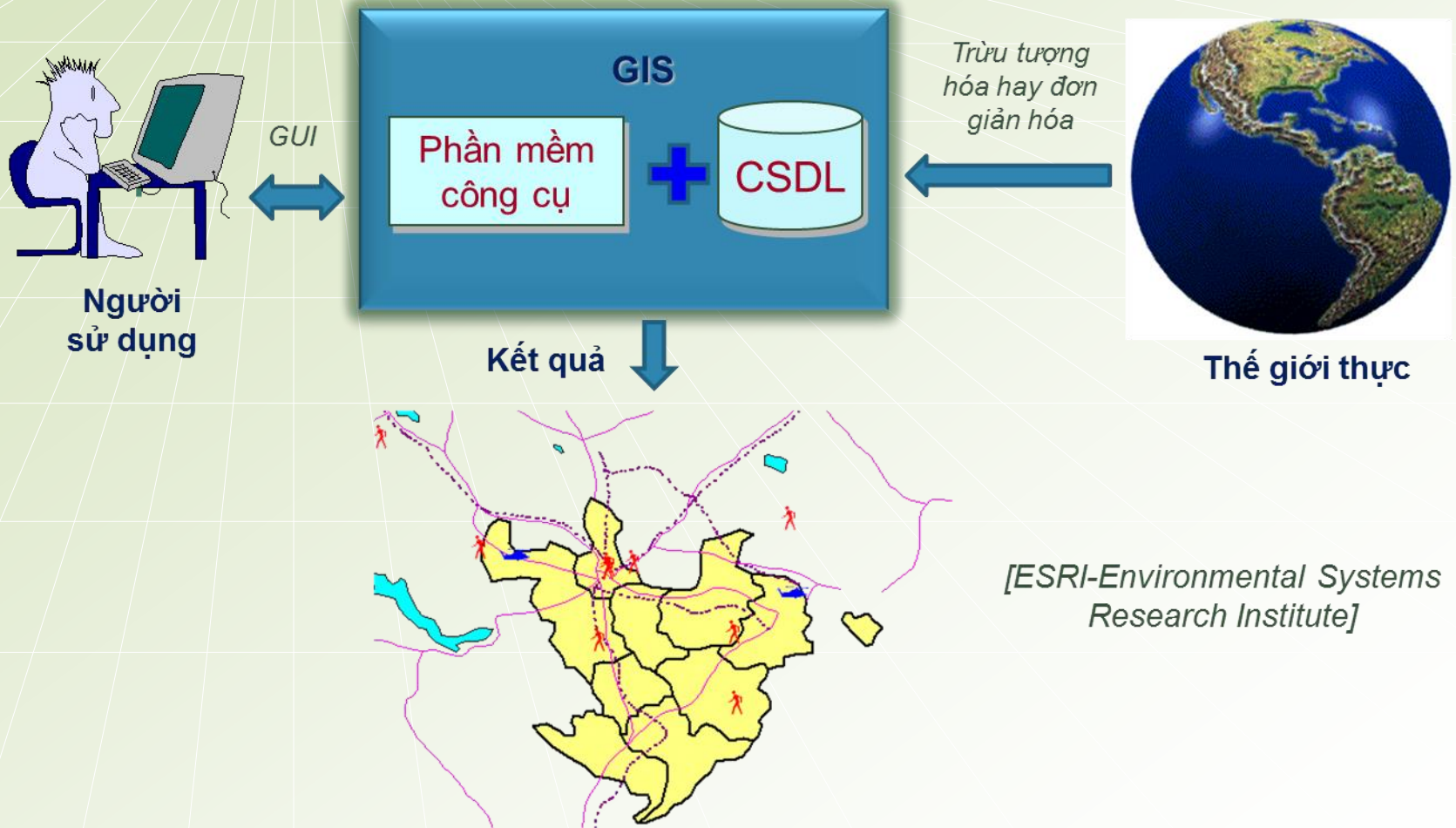
9



# Nội dung

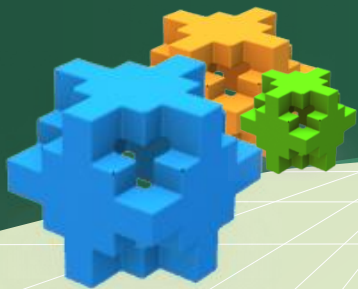
- ❖ Hiện tượng địa lý
- ❖ Biểu diễn đặc trưng địa lý
- ❖ Chỉ mục dữ liệu không gian
- ❖ Tổng kết bài

# Nhắc lại kiến thức Bài 1



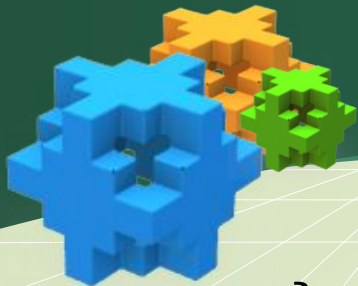
[ESRI-Environmental Systems Research Institute]





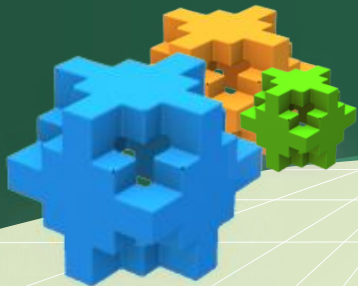
# 1. Hiện tượng địa lý

- ❖ GIS hoạt động với giả thiết các hiện tượng địa lý xảy ra trong không gian Euclid 2-D hoặc 3-D
- ❖ Hiện tượng hay tiến trình địa lý là các thực thể trong thế giới thực với khả năng:
  - Đặt tên hay mô tả được,
  - Tham chiếu địa lý được và
  - Được gán cho thời gian/khoảng thời gian mà nó tồn tại
- ❖ Hiện tượng địa lý được xác định bởi bộ ba (*description, georeference, time-interval*)
- ❖ Ví dụ: các đối tượng nghiên cứu của cơ quan địa chính là thửa ruộng, đường phố, đất thổ cư...
- ❖ Ví dụ khác?



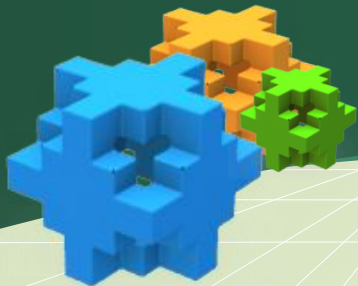
# Biểu diễn hiện tượng địa lý

- ❖ Để biểu diễn (mô hình hóa) được một phần thế giới thực (hiện tượng địa lý), ta cần phải trả lời các câu hỏi:
  - Biểu diễn cái gì?
  - Mức độ chi tiết đến đâu? và
  - Chu kỳ thời gian nào?
- ❖ Biểu diễn phải đủ thông tin, giảm dung lượng lưu trữ và tăng tốc độ tính toán để GIS có thể phân tích và con người có thể hiểu chúng rõ ràng hơn
- ❖ Biểu diễn trên cơ sở các quy tắc/luật, ví dụ:
  - Xấp xỉ không gian trên cơ sở luật Waldo Tobler “*Everything is related to everything else, but near things are more related than distant things*”
  - Sử dụng các phương pháp khái quát hóa (*generalization*) để loại bỏ những chi tiết không cần thiết.



# Các loại hiện tượng địa lý

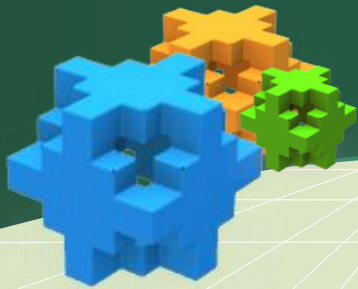
- ❖ Xác định và biểu diễn hiện tượng địa lý là công việc khó khăn vì chúng đa dạng -> cần phân loại
- ❖ Hai giả thiết ban đầu
  - Cần gán tên/mô tả và tham chiếu địa lý cho đối tượng địa lý, bỏ qua thời gian (nghiên cứu sau)
  - Một số hiện tượng địa lý xảy ra mọi nơi (*field*) trong khi một số khác chỉ xảy ra tại một số vị trí nhất định (*object*) của vùng nghiên cứu



# Các loại hiện tượng địa lý

- ❖ Quy tắc thông thường
  - Hiện tượng địa lý tự nhiên thuộc loại '*geographic field*'
  - Hiện tượng địa lý nhân tạo thuộc loại '*geographic object*'
- ❖ Nhiều trường hợp đặc biệt và cần được xem xét thận trọng khi áp dụng phân loại
- ❖ Ví dụ hiện tượng địa lý
  - The temperature at local noon on December 2<sup>nd</sup>, 2014 at latitude 34 degrees 45 minutes north, longitude 120 degrees 0 minutes west, was 18 degrees Celsius.
  - Mount Everest is 8848 m high.
  - Ví dụ khác?





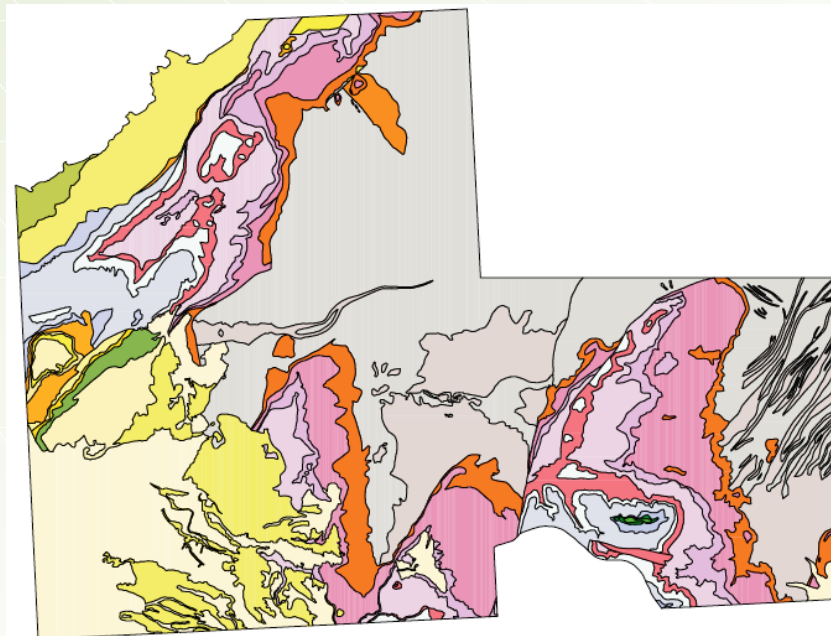
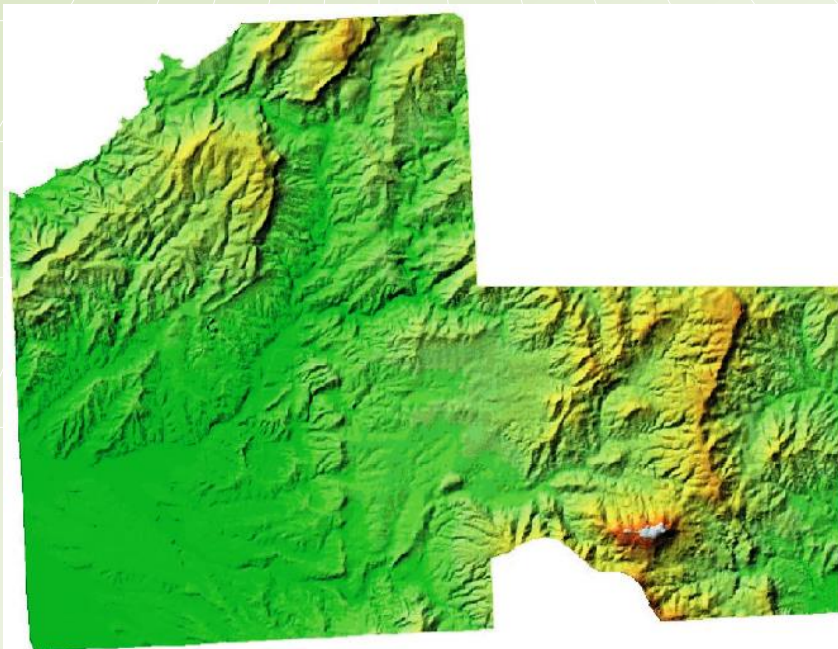
# Geographic Fields

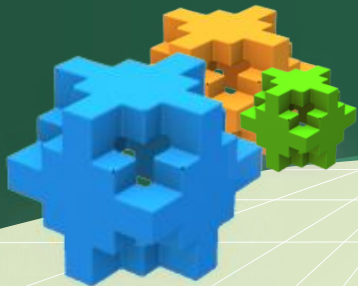
- ❖ Trường (*field*) là hiện tượng địa lý có giá trị “mọi nơi” trong vùng nghiên cứu
  - Trường được xem như hàm toán học  $f(x, y)$ , trong đó  $(x, y)$  là vị trí.
- ❖ Trường có thể là liên tục hoặc rời rạc
  - Trường liên tục:
    - Giá trị trong vùng nghiên cứu thay đổi từ từ
      - Ví dụ: Nhiệt độ không khí, áp suất, độ cao địa hình...
    - Có thể đạo hàm: Xác định mức độ thay đổi giá trị trường trên đơn vị khoảng cách ở bất kỳ đâu hay hướng nào
      - Ví dụ: Trường độ cao
  - Trường rời rạc:
    - Chia không gian nghiên cứu thành các phần không chồng chéo, có đường bao và có cùng giá trị
      - Ví dụ: Phân lớp đất, vùng thu hoạch...



# Ví dụ trường địa lý

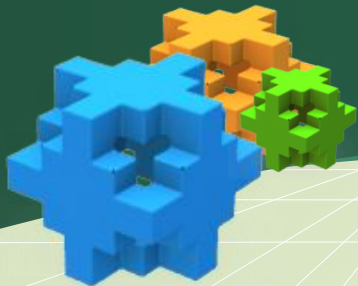
- ❖ Vùng nghiên cứu: *Falset, Spain*
- ❖ Trường liên tục biểu diễn độ cao
- ❖ Trường rời rạc biểu diễn các vùng địa chất khác nhau





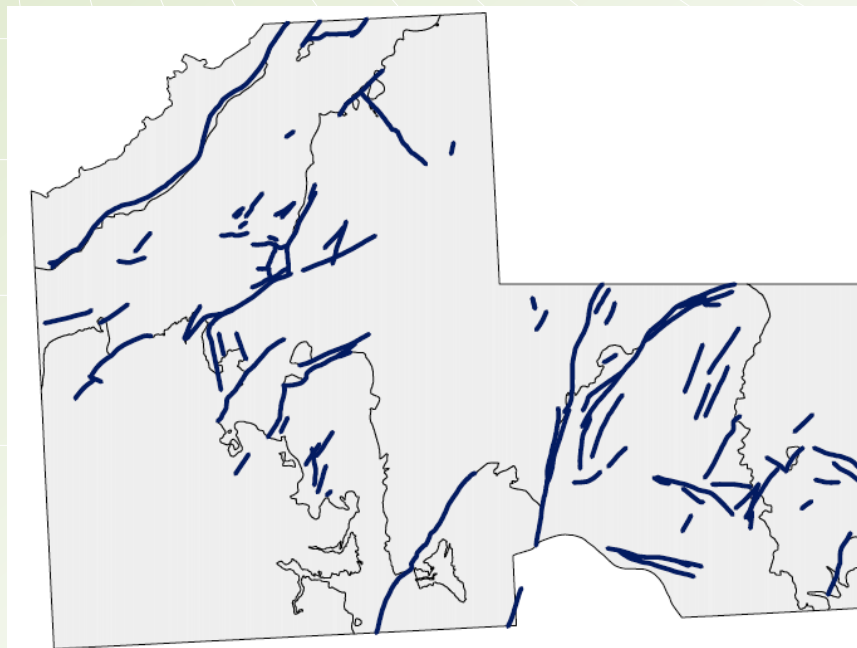
# Đối tượng địa lý

- ❖ *Geographic objects*: dễ dàng phân biệt với nhau và được đặt tên.
- ❖ Vị trí của đối tượng được xác định bởi tổ hợp các tham số sau:
  - Location (where is it?)
  - Shape (what form is it?)
  - Size (how big is it?)
  - Orientation (in which direction is it facing?)
- ❖ Ví dụ:
  - Trạm bán xăng (vị trí là quan trọng)
  - Đường quốc lộ (Vị trí: bắt đầu, kết thúc ở đâu; Hình dáng: Bao nhiêu làn đường; Hướng: Có thể đi về đâu)

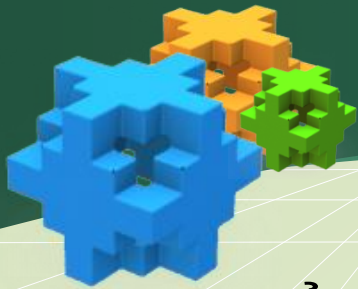


# Ví dụ đối tượng địa lý

- ❖ Biểu diễn đứt gãy địa chất (màu xanh) của vùng *Falset, Spain*
  - Mỗi đoạn đứt gãy đều có vị trí, hình dạng là đối tượng 1-D, kích thước là độ dài của đoạn, hướng không quan trọng.







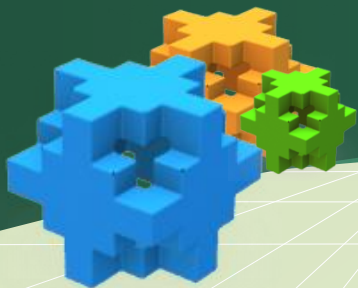
# Đối tượng địa lý

## ❖ Biểu diễn hình dạng:

- Đối tượng được biểu diễn bởi hình dạng 2-D (vùng, đa giác), ví dụ: Đơn vị hành chính, thửa đất, ao, hồ...
- Đối tượng được biểu diễn bởi dạng tuyến 1-D, ví dụ: Sông, ngòi, đường quốc lộ, đường phố, đường dây tải điện...
- Đối tượng được biểu diễn bởi điểm (0-D), ví dụ: các tiện ích, công trình công cộng, động vật...

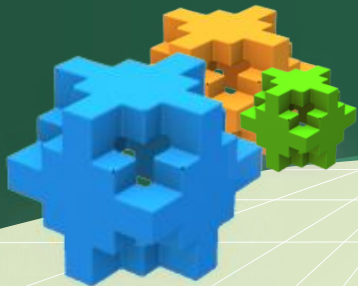
## ❖ Thông thường, cần phải nghiên cứu đồng thời các đối tượng địa lý (tập hợp các đối tượng thành đơn vị) để trả lời các câu hỏi, ví dụ như sau:

- Những đoạn đường nào cách trạm xăng 5 km?
- Đường đi ngắn nhất giữa hai thành phố?
- Bao nhiêu xe ô tô có thể đi từ thành phố A đến thành phố B trong 1 giờ?



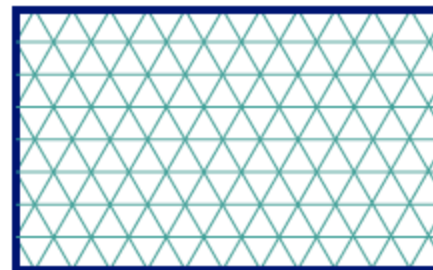
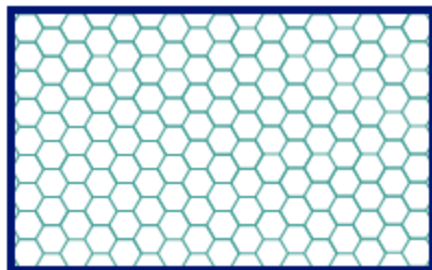
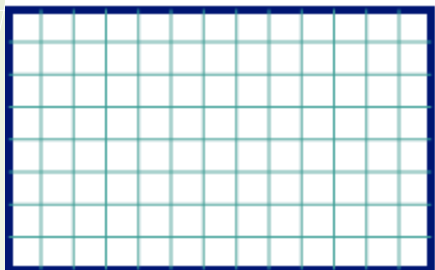
## 2. Biểu diễn đặc trưng địa lý

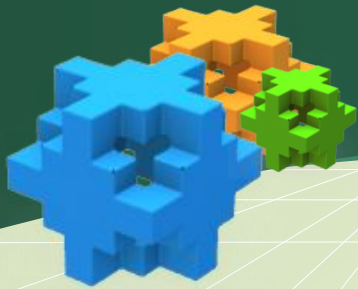
- ❖ Phần này nghiên cứu cách biểu diễn fields và objects trong máy tính sao cho hiệu quả
- ❖ Các hiện tượng địa lý thường có tính chất của hàm liên tục trong không gian, ví dụ địa hình.
- ❖ Để lưu trữ địa hình trong máy tính, có thể:
  - Lưu trữ nhiều nhất có thể cặp (*location, elevation*), hoặc
  - Tìm hàm biểu diễn trường địa hình,
    - Ví dụ: Hàm  $3.0678x^2 + 20.08x - 7.34y$ , với khả năng tính toán độ cao tại bất kỳ vị trí (x, y) nào.
- ❖ Nhận xét:
  - Cách 1: Không thể lưu trữ mọi giá trị độ cao cho mọi vị trí
  - Cách 2: Khó tìm ra được hàm phù hợp với vùng lớn
  - Trong GIS: Sử dụng kết hợp hai cách trên



# Kỹ thuật khảm đều

- ❖ Các fields thường được biểu diễn trong máy tính bởi tiệm cận *khảm* (*tessellation*);
- ❖ Các objects được biểu diễn bởi tiệm cận *vector*.
- ❖ Khảm là phân hoạch không gian thành các tế bào không chồng phủ nhau để tạo thành vùng nghiên cứu
  - Các tế bào có cùng kích thước và hình dạng
  - Gán thuộc tính cho các tế bào.
- ❖ Mỗi tế bào được kết hợp với giá trị thuộc tính
- ❖ Ba loại khảm đều: Hình vuông, lục lăng và tam giác





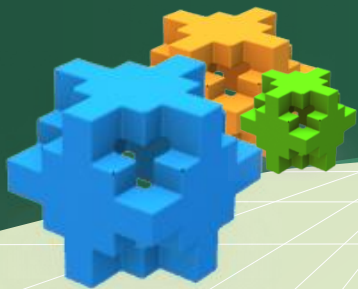
# Kỹ thuật khảm đều

- ❖ GIS sử dụng kỹ thuật khảm được gọi là *GIS raster*.

A *raster* is a set of regularly spaced (and contiguous) cells with associated (field) values. The associated values represent cell values, not point values. This means that the value for a cell is assumed to be valid for all locations within the cell.

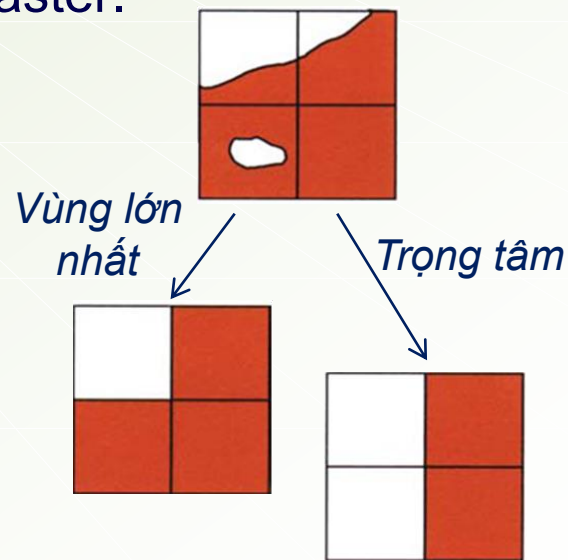
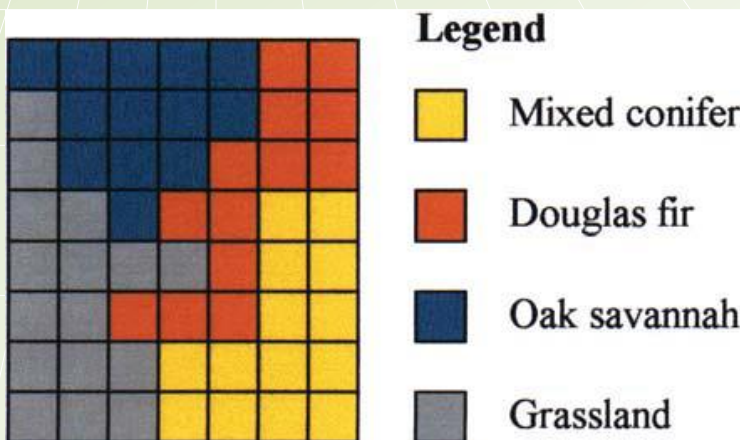
- ❖ Kích thước của vùng con mà một tế bào có thể biểu diễn được gọi là độ *phân giải raster*.
- ❖ Vị trí của tế bào lưu trữ là cố định, là điểm giữa của tế bào hay góc dưới trái của tế bào.
- ❖ Các vị trí khác được tính thông qua hàm nội suy
- ❖ Lợi thế của khảm đều là tính toán nhanh
- ❖ Nhược điểm: Không thích nghi với hiện tượng tự nhiên





# Kỹ thuật khảm đều

- ❖ Sự khác biệt về biểu diễn giá trị tế bào của hai trường địa lý (rời rạc và liên tục)
  - Trường rời rạc lưu trữ giá trị nguyên, còn gọi là 'integer' raster
    - Ví dụ loại đất sử dụng, loại cây trồng
  - Trường liên tục còn gọi là 'floating point' raster.
- ❖ Gán thuộc tính cho tế bào

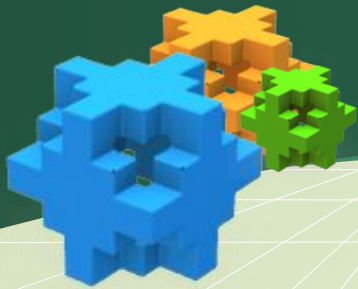


Kỹ thuật xác định giá trị thuộc tính của khảm đều



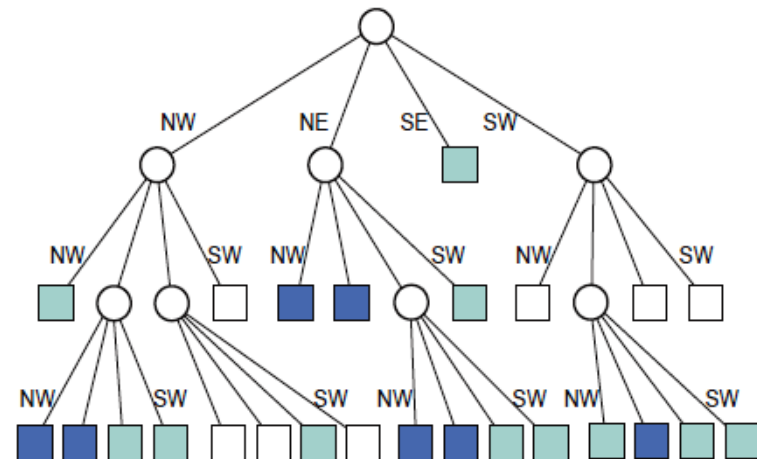
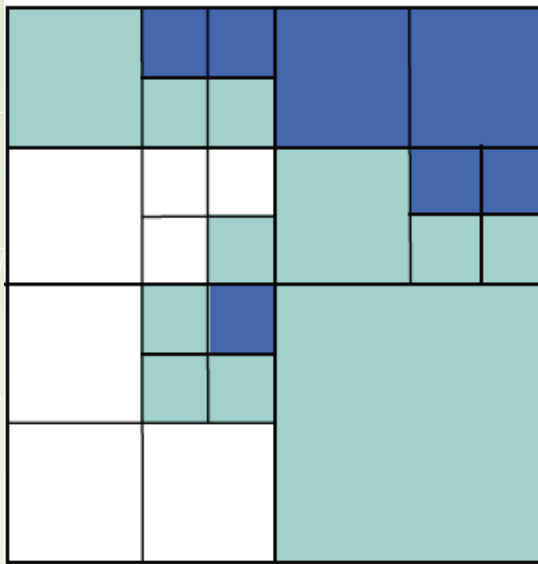
# Kỹ thuật khảm không đều

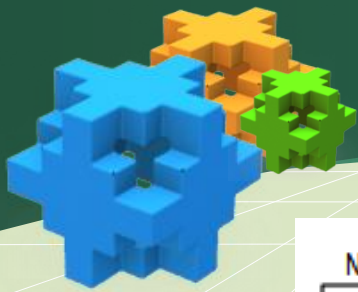
- ❖ *Irregular tessellations* là phân hoạch không gian thành các tế bào tách rời nhau, nhưng tế bào có thể có hình dạng và kích thước khác nhau
- ❖ Khảm không đều phức tạp hơn khảm đều, nhưng thích nghi tốt hơn với hiện tượng tự nhiên (ví dụ: bề mặt địa hình), dẫn tới tối ưu dung lượng lưu trữ
- ❖ Cấu trúc dữ liệu hay sử dụng: *Region Quadtree*
  - Trên cơ sở khảm đều với các tế bào hình vuông, nếu các tế bào láng giềng có cùng giá trị thuộc tính thì cùng biểu diễn bởi tế bào lớn hơn.



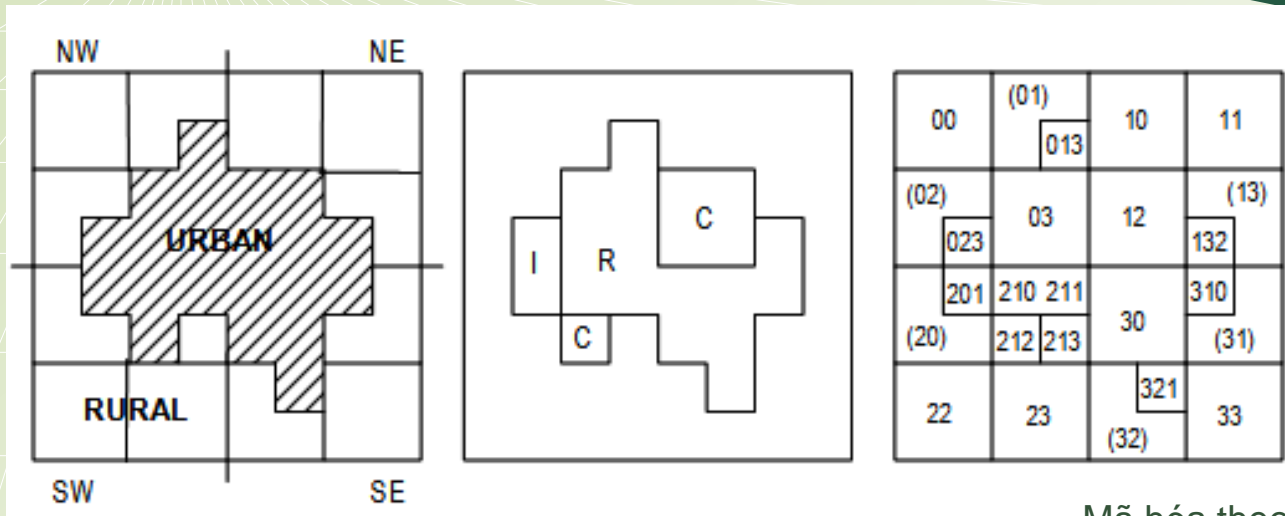
# Ví dụ khảm không đều

- ❖ Khảm sử dụng raster 8x8 với 3 giá trị field: white, green và blue.
- ❖ Cây tứ phân được hình thành bởi chia lặp vùng thành 4 phần: NW, NE, SE, SW

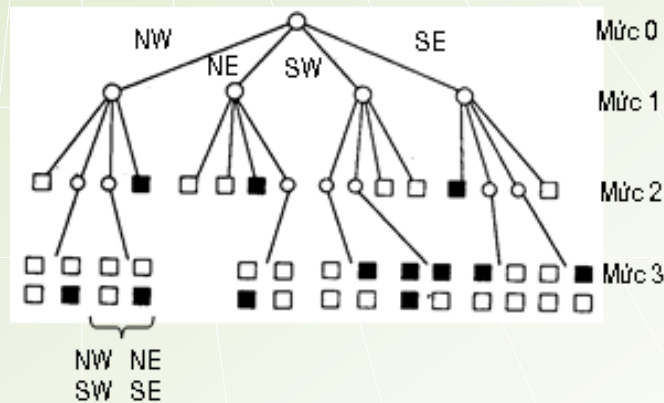




# Ví dụ Cây tứ phân vùng



Mã hóa theo Z



Mức cây			Loại đất
1	2	3	
		013	R
		023	I
	03		R
	12		C
		132	R
		201	I
		210	R
		211	R
		212	C
	30		R
		310	R

R – Thổ cư  
C - Thương mại  
I - Công nghiệp





# Kỹ thuật biểu diễn vector

- ❖ Biểu diễn vector: kết hợp tham chiếu địa lý với hiện tượng địa lý
- ❖ Tham chiếu địa lý là cặp tọa độ trong không gian địa lý, gọi là vector
- ❖ Có nhiều cách biểu diễn vector
  - Ví dụ mô hình TIN (*Triangulated Irregular Networks*) biểu diễn trường địa lý, được xem như kỹ thuật lai giữa khảm và biểu diễn vector.
  - Đây là cấu trúc dữ liệu hay được sử dụng nhất trong GIS



# Mô hình dữ liệu TIN

## ❖ Mô hình dữ liệu TIN

- TIN (*Triangulated Irregular Networks*) là kỹ thuật biểu diễn chuẩn cho mô hình số bề mặt

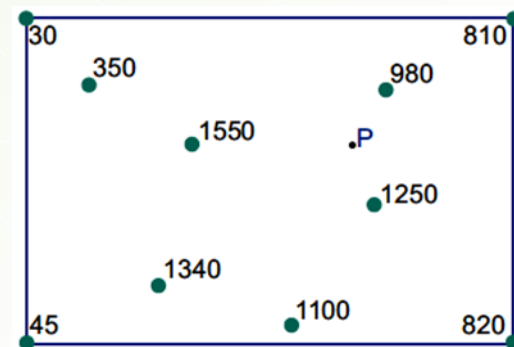
## ❖ Đầu vào: Tập vector với giá trị thuộc tính, ví dụ độ cao

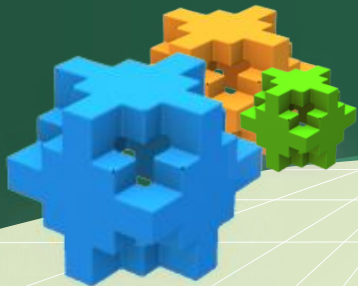
- Vị trí đo thuộc tính là tùy ý, không nhất thiết phân bố đều
- Được xem như điểm trong không gian 3D

## ❖ Nhiệm vụ: Xây dựng khảm không đều từ các tam giác

## ❖ Ý tưởng:

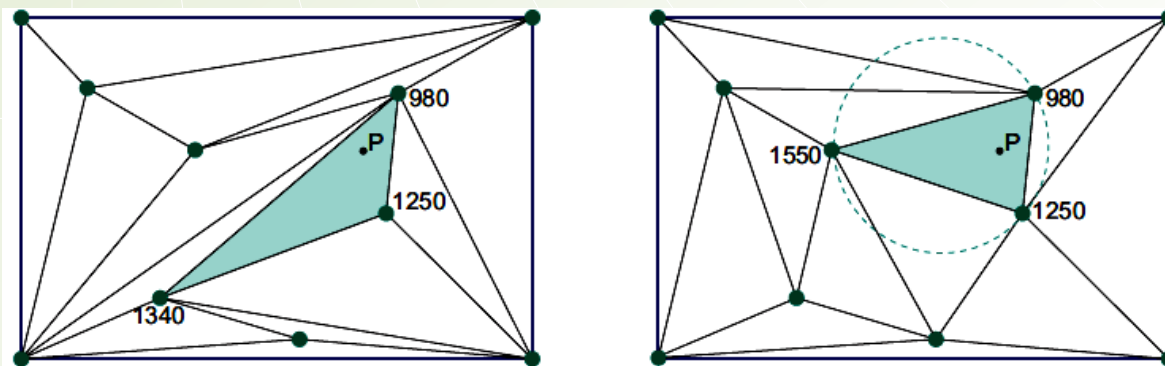
- Xác định 3 điểm thuộc TIN
- Xây dựng mặt phẳng từ 3 điểm, không cùng trên đường thẳng
- Từ phương trình mặt phẳng ta có thể xấp xỉ độ cao của các vị trí còn lại trong tam giác.





# Mô hình dữ liệu TIN

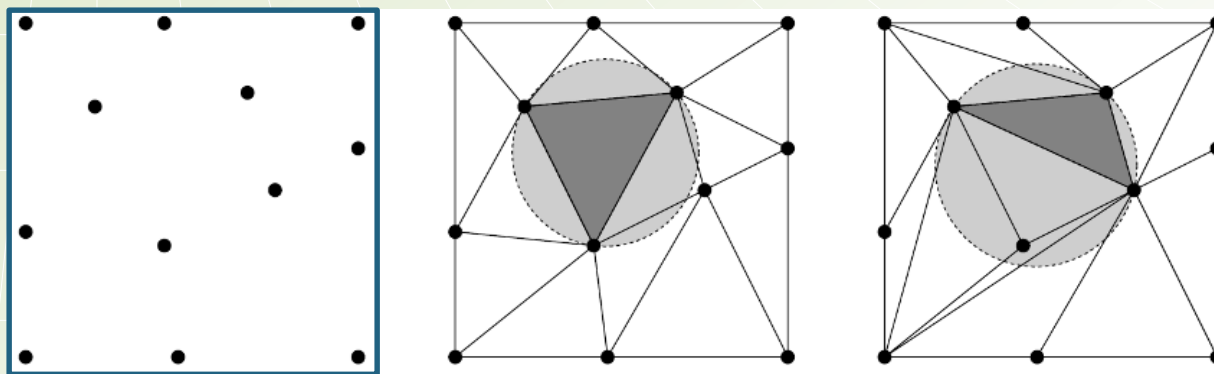
- ❖ Có thể xây dựng nhiều bộ tam giác từ tập vị trí đầu vào
  - Ví dụ có tập tam giác "gầy", và tập tam giác "đều hơn"
  - Từ tam giác ta có thể dễ dàng xấp xỉ giá trị cho các điểm khác, ví dụ P.
- ❖ Cần xác định tập tam giác nào là tốt nhất (có lỗi nhỏ nhất khi xấp xỉ độ cao)
  - Khi xấp xỉ P: cho lại giá trị khác nhau vì khoảng cách trung bình từ P đến các đỉnh tam giác là khác nhau (ở hai hình dưới)





# Mô hình dữ liệu TIN

- ❖ Bộ tam giác tốt nhất gọi là bộ tam giác thỏa mãn tiêu chí *Delaunay*.
- ❖ Hai tính chất cơ bản của *Delaunay*:
  - Tam giác có xu thế đều nhất có thể
  - Vòng tròn ngoại tiếp của tam giác không chứa bất kỳ tọa độ nào khác.



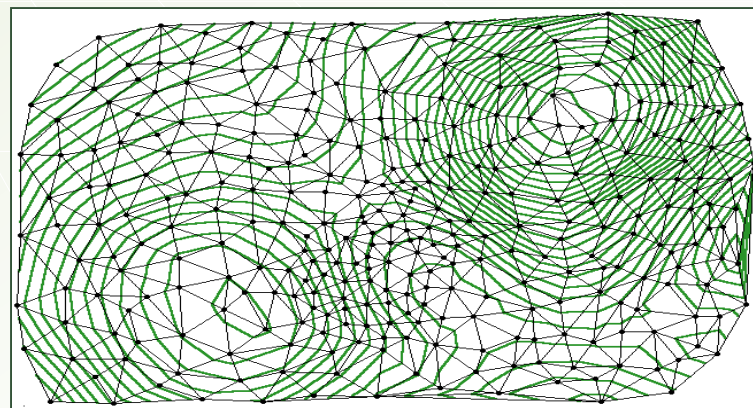
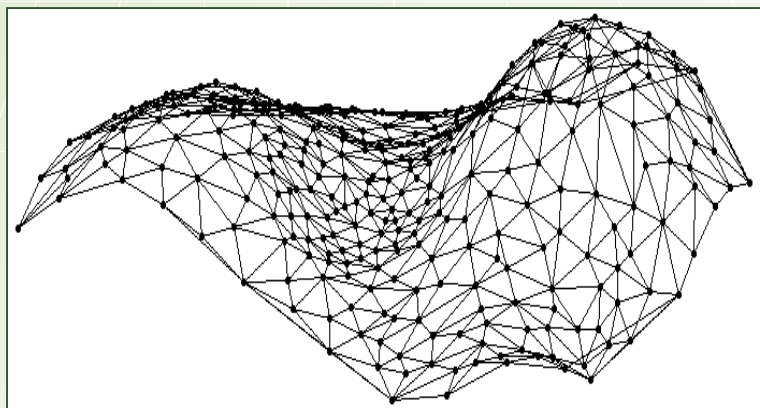
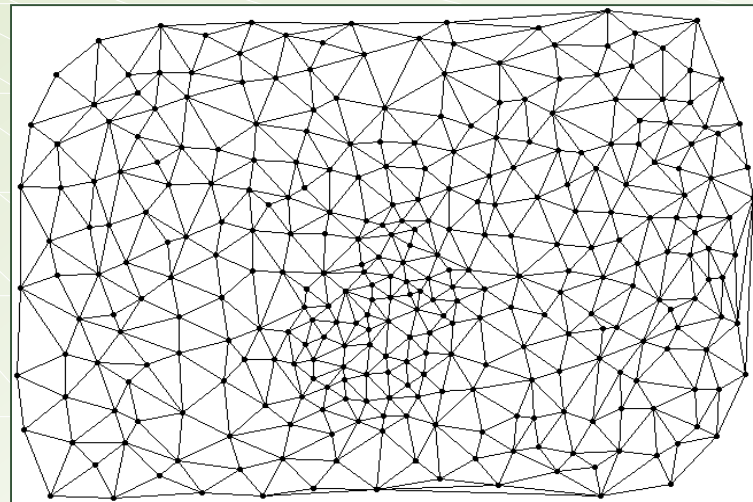
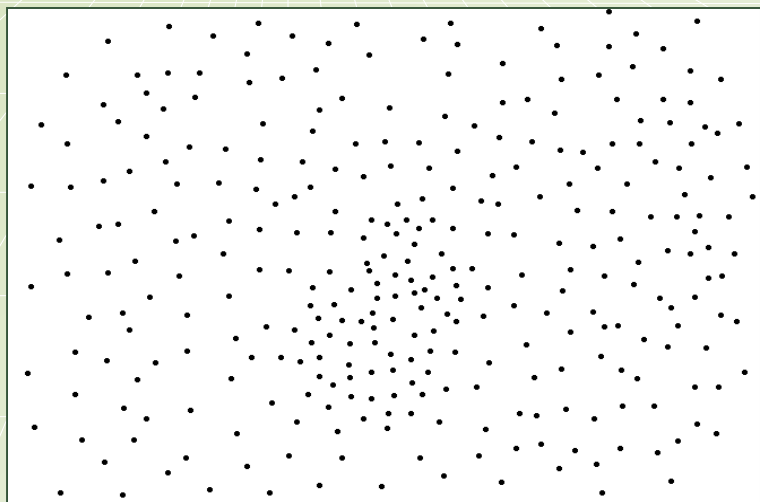
- ❖ TIN còn được gọi là khảm không đều. Các tế bào là các tam giác không đều, phủ vùng nghiên cứu.





# Mô hình dữ liệu TIN

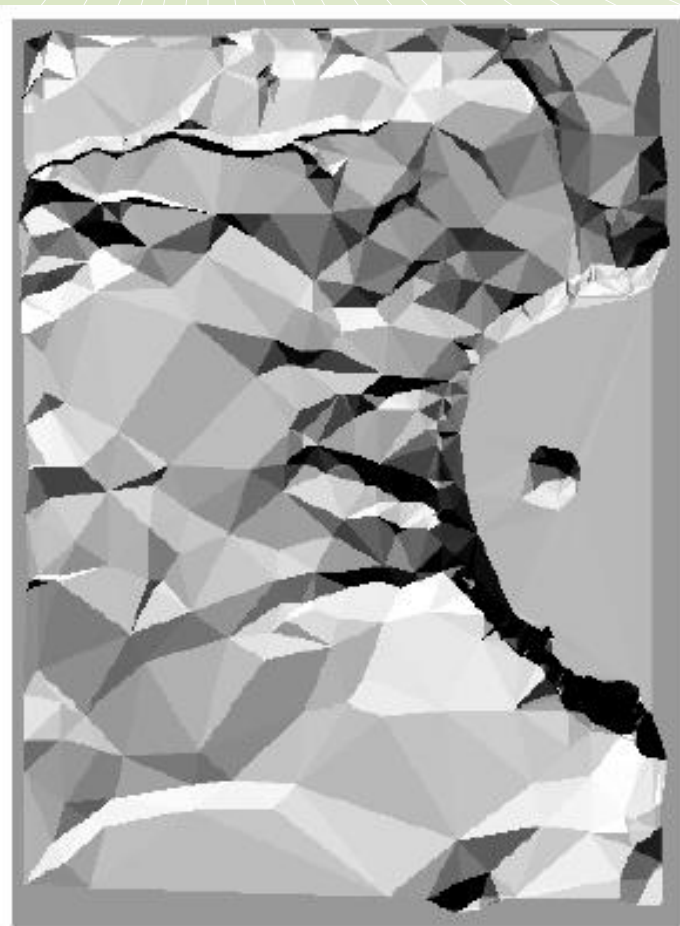
❖ Ví dụ



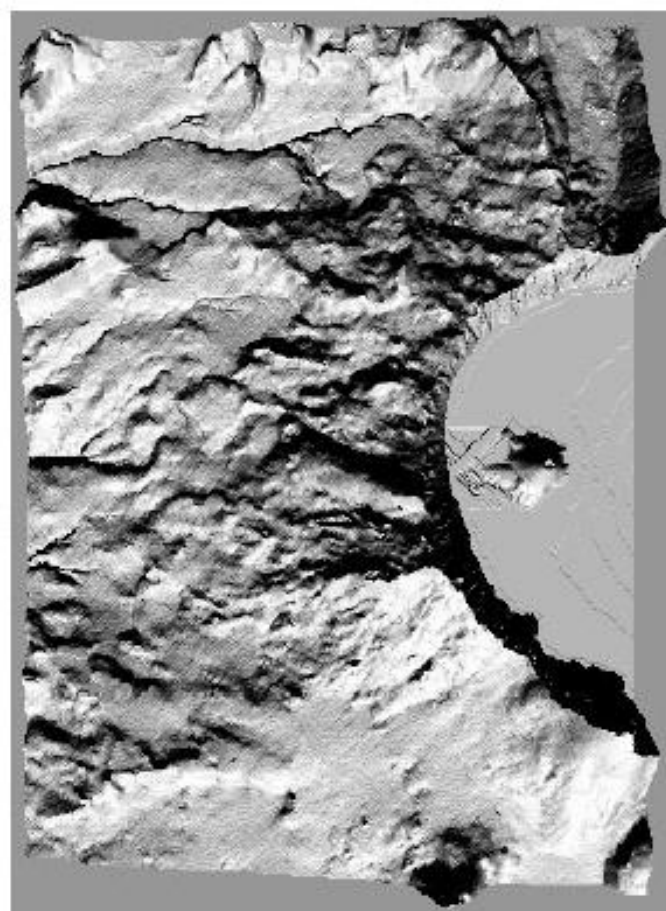


# Mô hình dữ liệu TIN

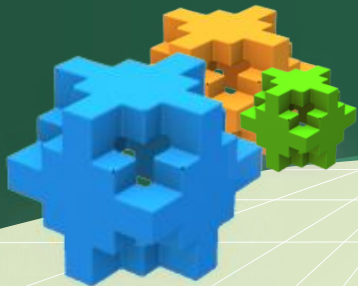
❖ Ví dụ



TIN với 771 đỉnh

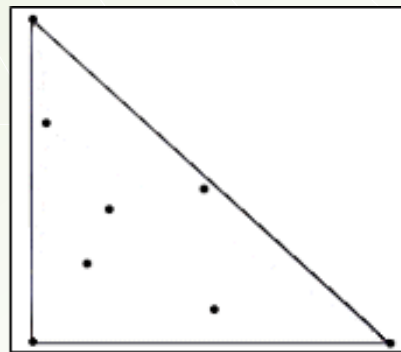
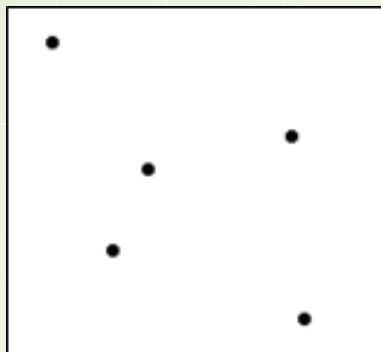


TIN với 154 221 đỉnh



# Thuật toán Norm

- ❖ Thuật toán xây dựng TIN từ các tập điểm do GS. Norm Jones ([Brigham Young University](#)) đề xuất
- ❖ Thuật toán
  - Input: Tập các điểm trong mặt phẳng Euclidean
  - Output: Mô hình TIN của các tập điểm
  - Bước 1: Loại bỏ các điểm trùng nhau
  - Bước 2: Tính giá trị x, y cực đại của các điểm để xây dựng tam giác bao toàn bộ các điểm



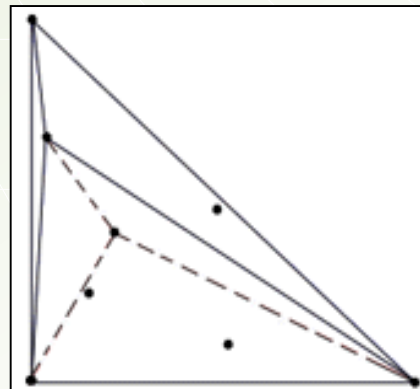
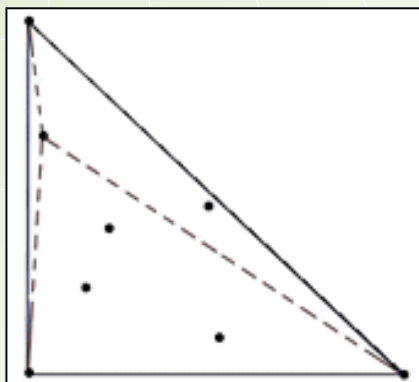




# Thuật toán Norm

## ❖ Thuật toán (tt)

- Bước 3: Kết nạp lần lượt các điểm vào TIN. Lặp các bước 4-6 cho đến khi toàn bộ các điểm đã được đưa vào TIN
- Bước 4: Với mỗi điểm vào, hãy tìm tam giác chứa chúng và gọi là tam giác vào. Nếu điểm nằm trên cạnh chung của 2 tam giác, lấy cả 2 tam giác làm tam giác vào
- Bước 5: Từ một điểm vào trong tam giác, chia tam giác thành 3 (hoặc 4 khi điểm nằm trên cạnh) tam giác nhỏ hơn



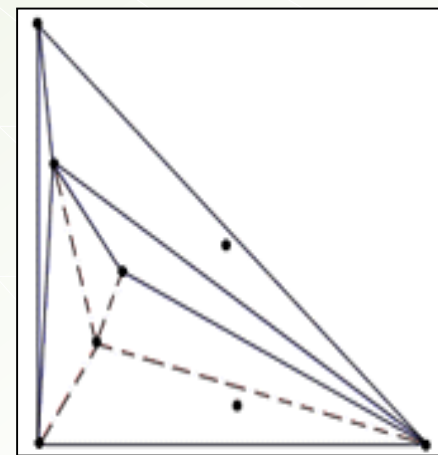
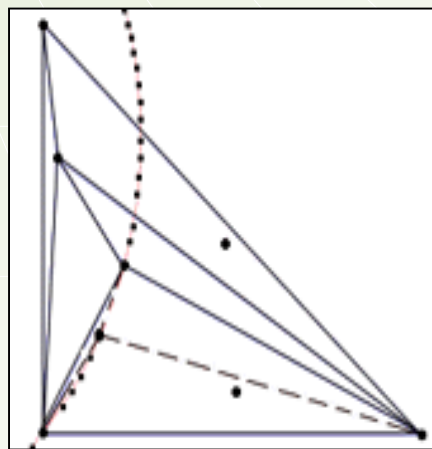
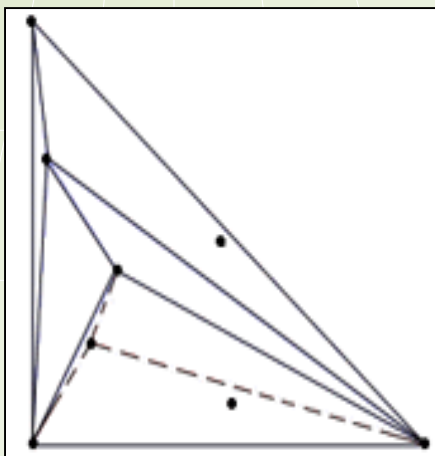




# Thuật toán Norm

## ❖ Thuật toán (tt)

- Bước 6: Kiểm tra mỗi tam giác nhỏ mới hình thành có thoả mãn điều kiện Delaunay hay không. Nếu thoả mãn chuyển sang bước 7, nếu không thoả mãn đổi chỗ các cạnh của các tam giác láng giềng (liền kề) và kiểm tra lại điều kiện Delaunay. Tiếp tục cho đến khi tất cả các tam giác thoả mãn điều kiện Delaunay.

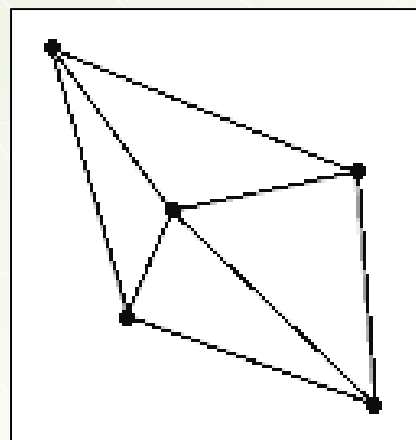
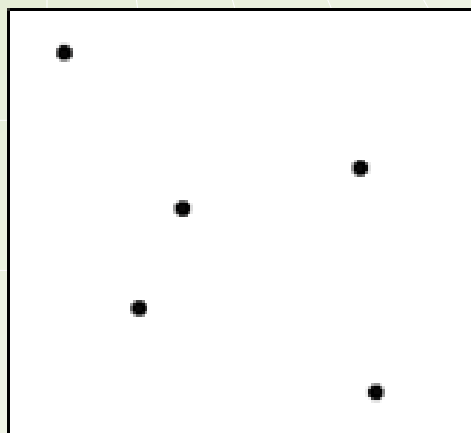


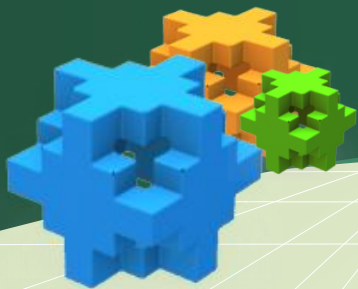


# Thuật toán Norm

## ❖ Thuật toán (tt)

- Bước 7: Lặp lại từ bước 4 đến bước 6 cho đến khi tất cả các điểm đầu vào được đưa vào lưới tam giác.
- Bước 8: Xoá tất cả các đỉnh của tam giác bao được tạo ra trong bước 2. Xoá tất cả các tam giác nối tới các đỉnh của tam giác bao và kết thúc.



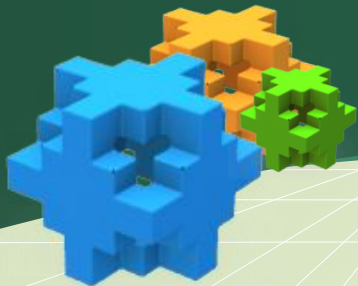


# Thuật toán Norm

## ❖ Kiểm tra điều kiện Delaunay

- Điểm  $D=(D_x, D_y)$  nằm trong vòng tròn đi qua ba điểm  $A=(A_x, A_y)$ ,  $B=(B_x, B_y)$ ,  $C=(C_x, C_y)$  hay không?
  - Sắp xếp A, B, C theo chiều kim đồng hồ
  - D nằm trong vòng tròn qua A, B, C khi và chỉ khi thỏa mãn:

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x - D_x & A_y - D_y & (A_x^2 - D_x^2) + (A_y^2 - D_y^2) \\ B_x - D_x & B_y - D_y & (B_x^2 - D_x^2) + (B_y^2 - D_y^2) \\ C_x - D_x & C_y - D_y & (C_x^2 - D_x^2) + (C_y^2 - D_y^2) \end{vmatrix} > 0$$



# Biểu diễn bằng véc tơ

## ❖ Biểu diễn đối tượng 'điểm'

- Điểm (*point*) được định nghĩa bởi bộ 2, tọa độ  $(x,y)$  trong 2D hay bộ ba, tọa độ  $(x,y,z)$  trong 3D
- Điểm được sử dụng để biểu diễn các đối tượng không có hình dạng và kích thước
- Ngoài **tham chiếu địa lý**, các dữ liệu kèm theo được lưu trữ với điểm là các **thuộc tính**
  - Ví dụ: Các tiện ích công cộng như ATM, bệnh viện, nhà hàng, bãi đỗ xe...
- Việc lựa chọn dữ liệu dạng điểm để biểu diễn đối tượng phụ thuộc vào ứng dụng cụ thể và tỷ lệ bản đồ.
  - Ví dụ Bệnh viện có thể là đối tượng điểm hoặc đối tượng 2D.

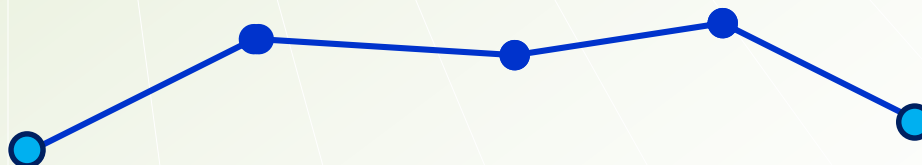




# Biểu diễn trong máy tính

## ❖ Biểu diễn đối tượng 'đường'

- Dữ liệu dạng đường (*line*) biểu diễn các đối tượng 1D như đường quốc lộ, tàu hỏa, sông ngòi, đường điện...
- Việc lựa chọn dữ liệu dạng đường để biểu diễn đối tượng phụ thuộc vào ứng dụng cụ thể và tỷ lệ bản đồ.
  - Ví dụ, đường phố có thể là đối tượng 2D hoặc 1D.
- Để biểu diễn một tuyến liên tục với dạng bất kỳ là khó khăn. GIS lựa chọn các điểm (nút) để lưu trữ và xấp xỉ đối tượng đường.
  - Line được định nghĩa bởi tọa độ hai đầu nút (*node*) và 0 hay nhiều nút bên trong (*vertex*)
- *Line* trong GIS còn được gọi là *polyline*, *arc* hay *edge*.

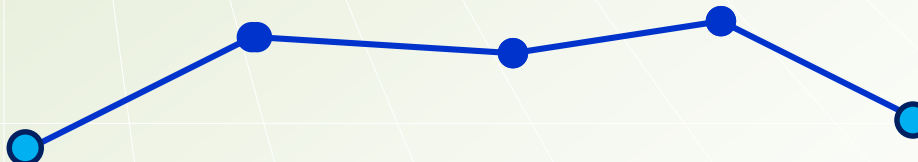


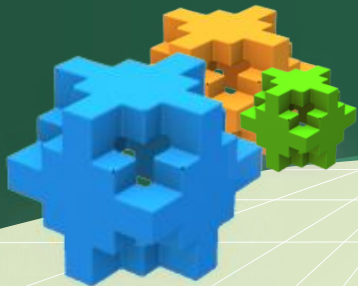


# Biểu diễn trong máy tính

## ❖ Biểu diễn đối tượng đường (tt)

- GIS lưu trữ các nút để hình thành *Line* theo các cách sau:
  1. Nối các nút bằng đoạn thẳng (*segment*) để hình thành đường gấp khúc (*polyline*).
  2. Biểu diễn đường cong bằng hàm tham số nối các nút.
- Hầu hết phần mềm GIS sử dụng giải pháp thứ 1
  - Để dễ biểu diễn hiện tượng tự nhiên
  - Tăng độ chính xác bằng bổ sung các *vertex*

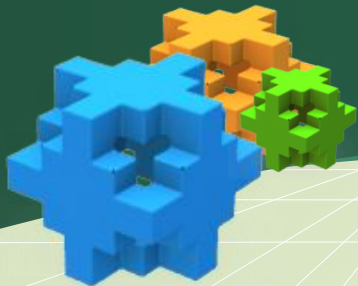




# Biểu diễn trong máy tính

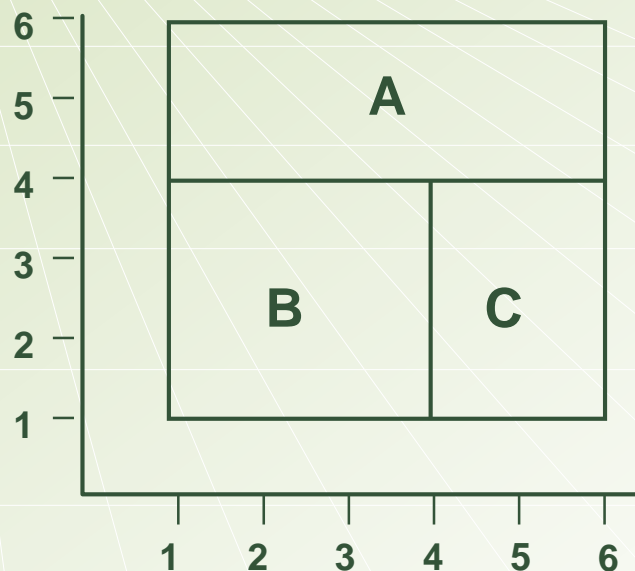
## ❖ Biểu diễn đối tượng dạng ‘vùng’

- Mỗi đặc trưng vùng được biểu diễn bởi cấu trúc cung/nút (là đường bao của vùng) để xác định đa giác.
- Các đặc trưng vùng cùng loại được lưu trữ trong cùng *layer* (được biểu diễn bởi các đa giác không xếp chồng lên nhau)
- Hai cấu trúc dữ liệu hay được sử dụng
  - *Spaghetti*
  - *Topology*



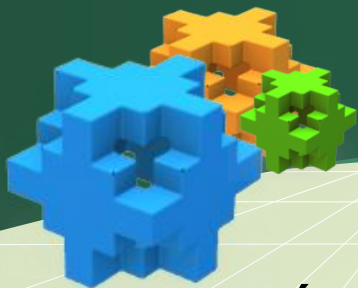
# Cấu trúc dữ liệu *spaghetti*

- ❖ Không có quan hệ giữa điểm, xâu và vùng.



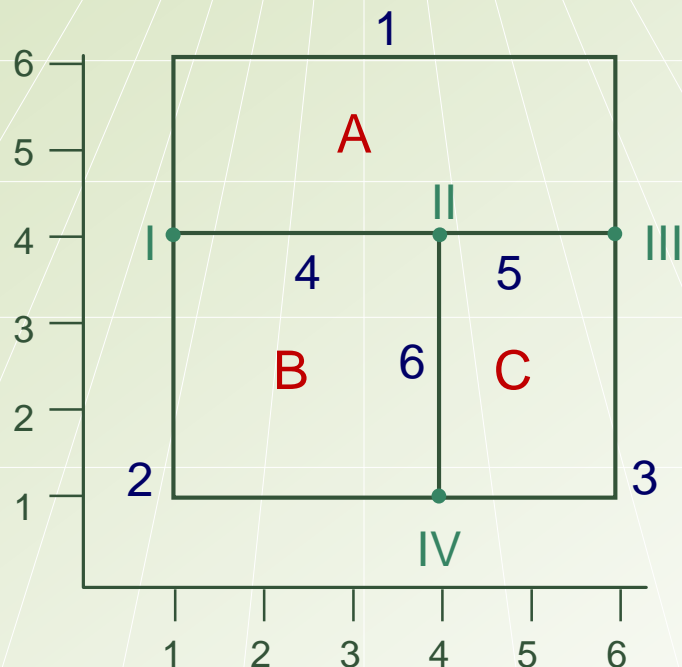
Area	Coordinates
A	(1,4), (1,6), (6,6), (6,4), (4,4), (1,4)
B	(1,4), (4,4), (4,1), (1,1), (1,4)
C	(4,4), (6,4), (6,1), (4,1), (4,4)





# Cấu trúc dữ liệu topo

## ❖ Cấu trúc dữ liệu topo theo mô hình cung-nút



Line	From	To	Left	Right	Vertexlist
1	I	III	O	A	
2	I	IV	B	O	
3	III	IV	O	C	
4	I	II	A	B	
5	II	III	A	C	
6	II	IV	C	B	

O = "outside" polygon

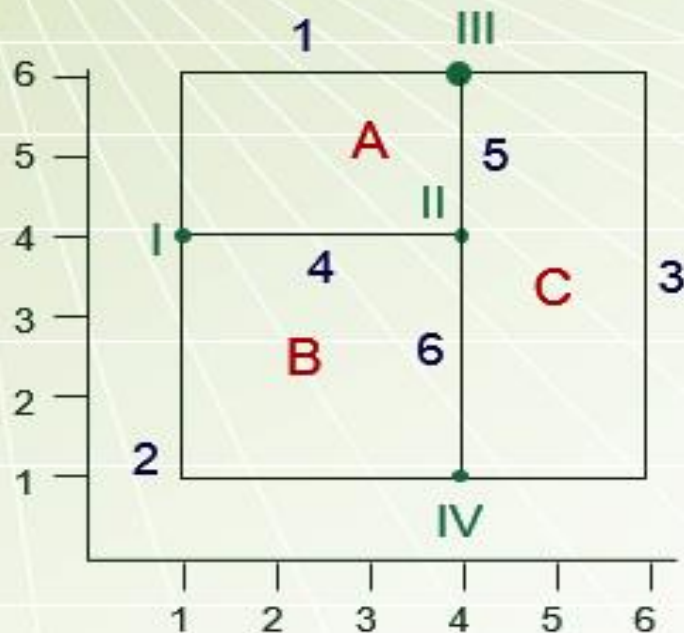
Node	X	Y
I	1	4
II	4	4
III	6	4
IV	4	1

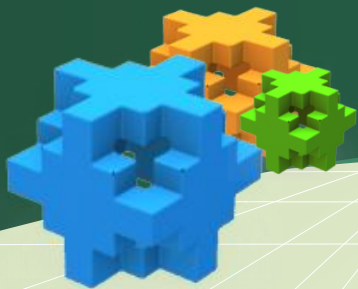
Poly	Lines
A	1,4,5
B	2,4,6
C	3,5,6



# Cấu trúc dữ liệu topo

- ❖ Bài tập: Xây dựng cấu trúc topo lưu trữ vùng nghiên cứu sau đây:

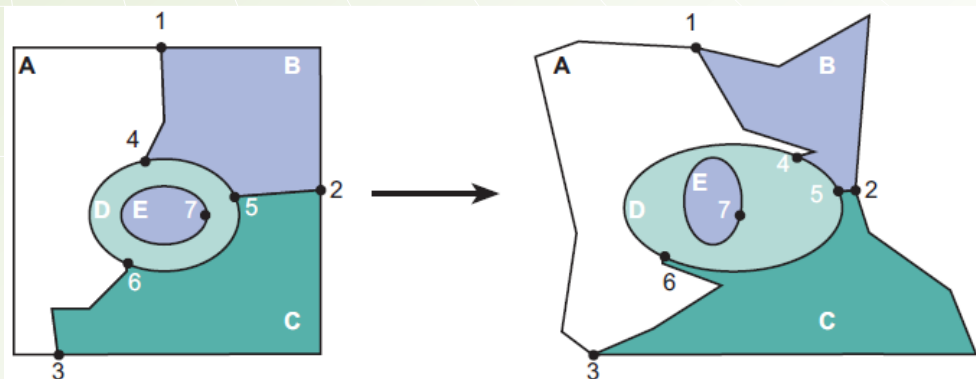


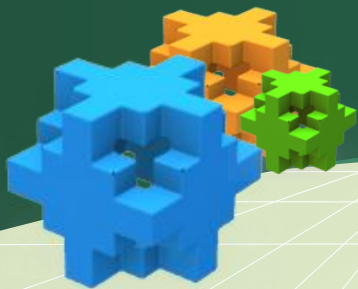


# Cấu trúc dữ liệu topo

## ❖ Các đặc trưng *topological*

- Topology đề cập đến các tính chất không gian bất biến dưới phép biến đổi nào đó
  - Ví dụ biến đổi “*tám cao su*”: Không gian biến đổi nhưng nhiều tính chất của các đặc trưng là bất biến
    - Vùng E vẫn trong vùng D
    - Các quan hệ láng giềng của A,B,C,D và E không thay đổi
    - Các vùng vẫn được bao bởi đường biên, chỉ có hình dạng và bán kính thay đổi.

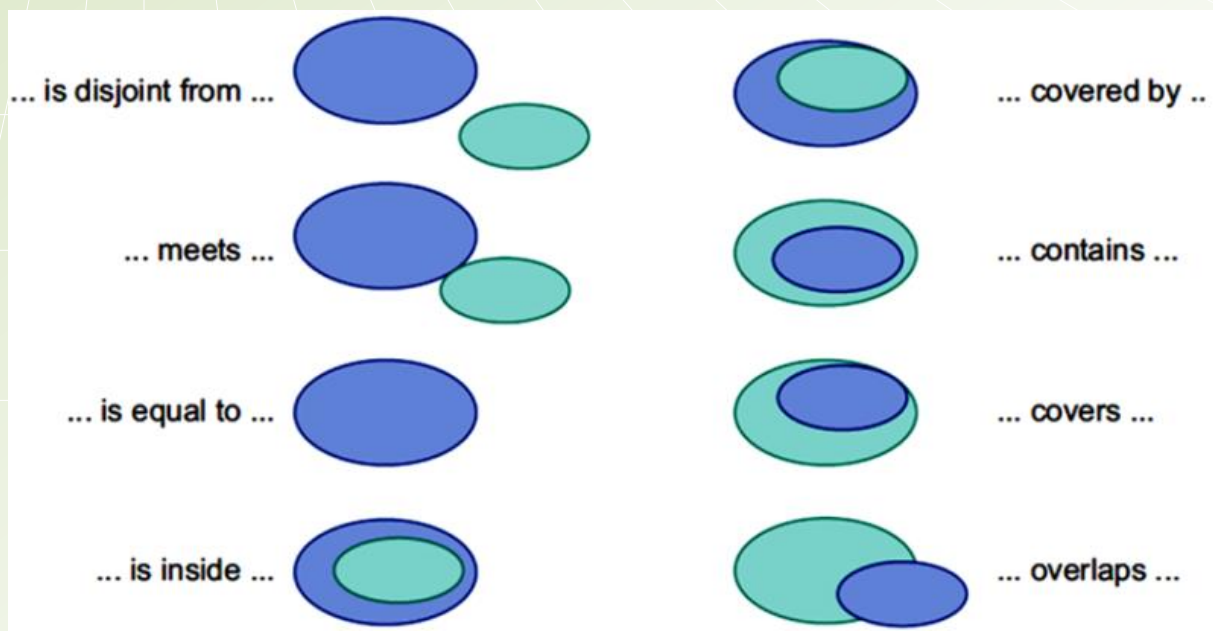




# Cấu trúc dữ liệu topo

## ❖ Topology và quan hệ không gian

- Có thể sử dụng tính chất topology như *interior* và *boundary* để định nghĩa quan hệ giữa các đặc trưng
- Từ bất biến topo giao cắt (*intersection*) giữa *interior* và *boundary* ta có các quan hệ không gian giữa hai vùng







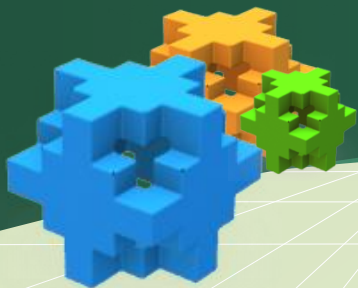
# Cấu trúc dữ liệu topo

## ❖ Topology và quan hệ không gian

- Ví dụ định nghĩa toán học của quan hệ *meets* như sau:

$$\begin{aligned} A \text{ meets } B &\stackrel{\text{def}}{=} \text{interior}(A) \cap \text{interior}(B) = \emptyset \wedge \\ &\text{boundary}(A) \cap \text{boundary}(B) \neq \emptyset \wedge \\ &\text{interior}(A) \cap \text{boundary}(B) = \emptyset \wedge \\ &\text{boundary}(A) \cap \text{interior}(B) = \emptyset \end{aligned}$$

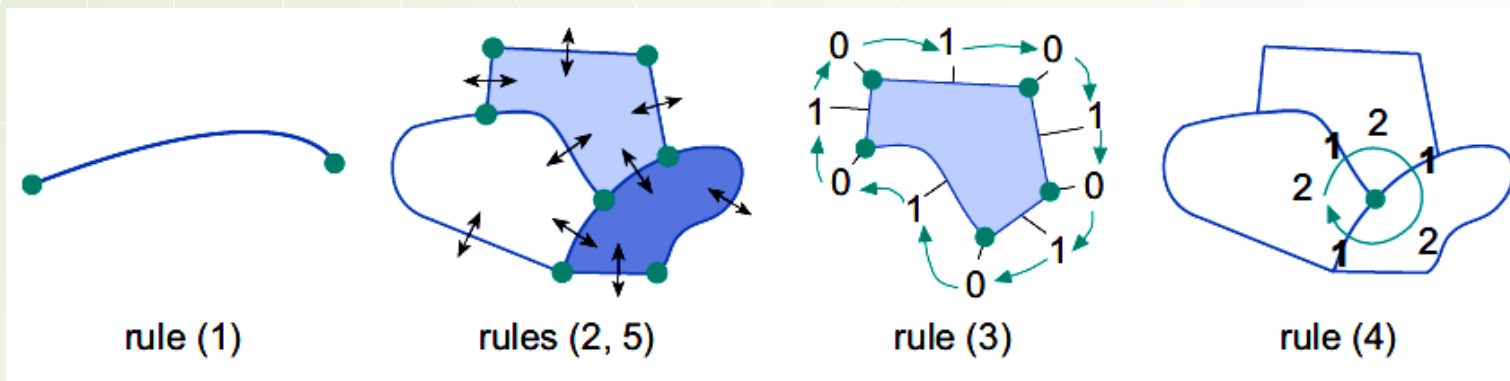
- Sử dụng các quan hệ *disjoint, meets, equals, inside, covered by, contains, covers, overlaps* để truy vấn dữ liệu trong CSDL không gian.

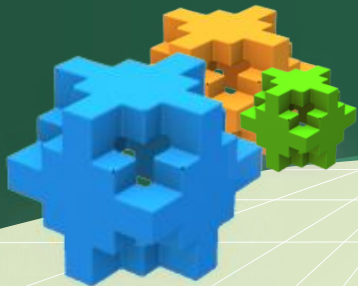


# Cấu trúc dữ liệu topo

## ❖ Topology và quan hệ không gian

- Năm qui tắc nhất quán topo trong không gian 2D
  1. Mỗi cung (*arc*) phải được bao bởi 2 nút (hai đầu mút)
  2. Mỗi cung liền kề với 2 đa giác (đa giác “trái”, “phải”)
  3. Mỗi đa giác (*polygon*) được bao kín bởi dãy lần lượt các cung, nút
  4. Xung quang nút (*node*) tồn tại lần lượt cung, đa giác
  5. Cung chỉ đi qua các nút của chúng.
- Sử dụng 05 qui tắc trên vào việc xây dựng đối tượng địa lý trong CSDL không gian.





# Cấu trúc dữ liệu topo

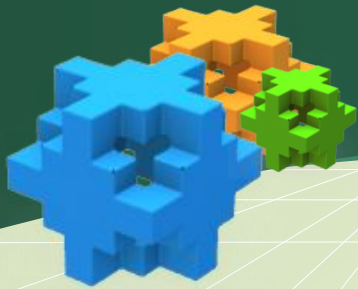
## ❖ Các đặc trưng *topological*

### ■ Ưu điểm

- Người sử dụng dễ dàng nhìn thấy quan hệ topo lưu trữ ngay trong CSDL
- Dữ liệu hình học chỉ cần lưu trữ một lần để tối ưu kích thước CSDL và loại bỏ các *slivers* khi số hóa
- Dễ dàng xâm nhập thông qua SQL API

### ■ Nhược điểm

- Hiệu năng truy vấn thấp: Phải tổ hợp nhiều bảng dữ liệu
- Kiểm tra toàn vẹn dữ liệu khó khăn
- Hiệu năng cập nhật thấp
- Cập nhật phức tạp.



## 4. Chỉ mục dữ liệu không gian

- ❖ CSDL địa lý thường là rất lớn, tốn kém thời gian truy vấn địa lý trong CSDL
  - Ví dụ tìm kiếm khách hàng (points) ở trong vùng (polygon) thương mại thường tốn kém thời gian. Có thể từ 10-100s trong CSDL với 50 triệu khách hàng.
- ❖ Giải pháp nâng cao tốc độ truy vấn
  - Cấu trúc topology
    - ví dụ: adjacency và network connectivity
  - Chỉ mục dữ liệu trong CSDL
    - Sử dụng index để tìm ra các bản ghi
    - Cho phép tìm kiếm ngẫu nhiên, không theo trình tự





# Chỉ mục dữ liệu

❖ Chỉ mục CSDL là trình diễn thông tin đặc biệt về đối tượng để nâng cao hiệu năng tìm kiếm.

- Ví dụ chỉ mục B-tree (*Balanced Tree*) một chiều trong hầu hết DBMS thương mại
  - Ví dụ tìm kiếm giá trị 72 chỉ cần 6 phép thử trong B-Tree
- B-tree không phù hợp với dữ liệu không gian, có các phương pháp chỉ mục khác hiệu quả hơn.

Original Data	B-Tree Indexed Data		
	Level 1	Level 2	Level 3
1	36	22	1
13			13
69			14
52			22
25		36	25
26			26
71			31
36			36
22		68	52
72			53
67			67
68			68
14			69
70			70
31			71
53			72



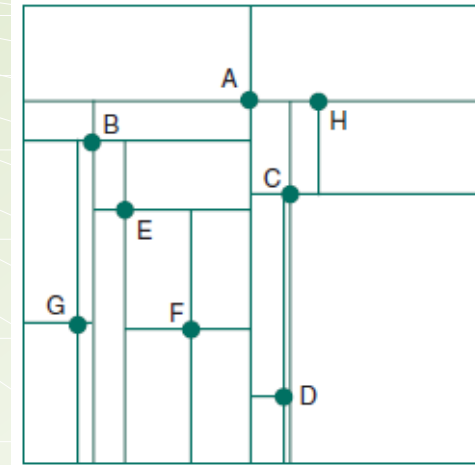
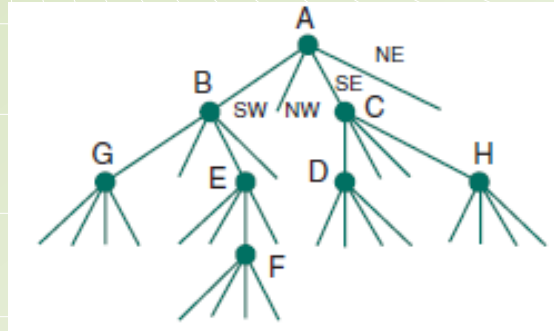
# Cây tứ phân

- ❖ *Quadtree* được hình thành bởi phân chia đệ qui không gian layer đặc trưng thành 4 phần phụ thuộc vào mật độ dữ liệu
- ❖ Phân loại *Quadtree* theo
  - Loại dữ liệu chỉ mục (điểm, đường, vùng, mặt, raster)
  - Thuật toán phân chia layer
  - Phân chia theo mật độ cố định hay thay đổi
- ❖ Ví dụ loại cây tứ phân
  - Point Quadtree
  - Region Quadtree

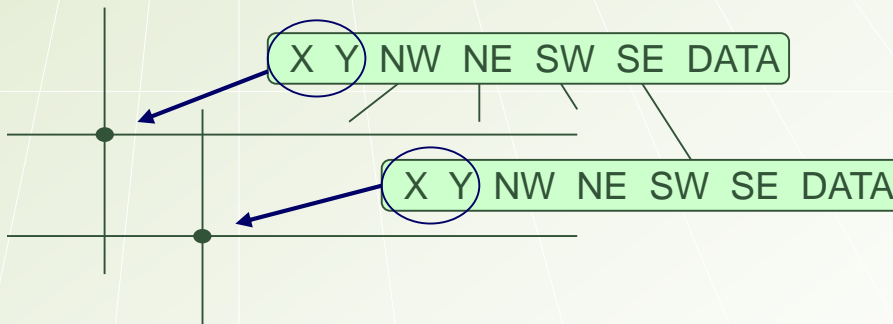


# Point Quadtree

## ❖ Ví dụ cây tứ phân điểm

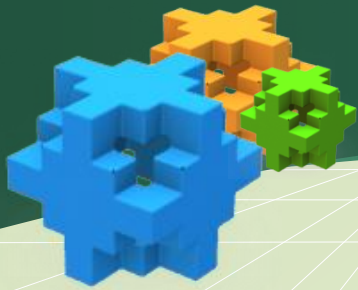


- Nút không phải là lá có hai trường lưu vị trí điểm, dữ liệu thuộc tính và 4 con trỏ trỏ đến các nhánh NW, NE, SW, SE
- Vị trí phân hoạch mặt phẳng thành 4 tại chính tâm điểm dữ liệu



```

qtnodetype=record
  INFO: infotype;
  XVAL: real;
  YVAL: real;
  NW, SW, NE, SE: ↑qtnodetype
end
  
```



# Point Quadtree

## Quadtree insert algorithm

**Input:** Given a point  $p$  and a quadtree  $Q$

**Output:** The quadtree  $Q$  updated by the insertion of the point  $p$

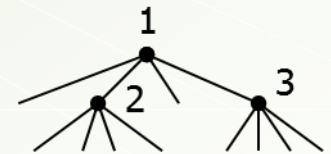
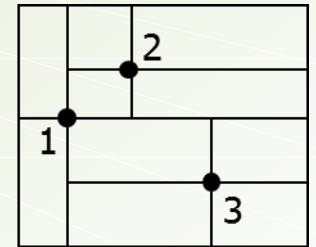
### Procedure:

Step 1. if  $Q$  is a null tree with no nodes

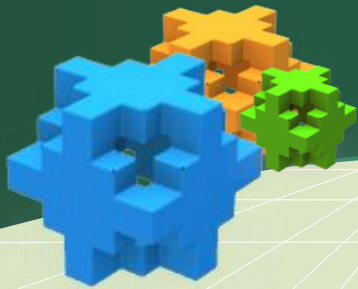
Step 2. then make a quadtree with root  $p$  and null tree as its four descendants

Step 3. else use the *quadtree branch algorithm* to evaluate the appropriate subtree

Step 4. recursively apply the algorithm to insert  $p$  into that subtree.







# Point Quadtree

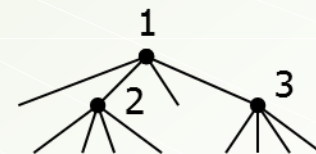
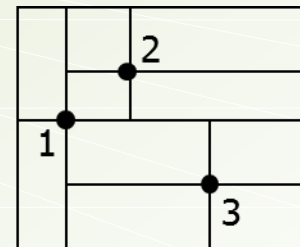
## Quadtree branch algorithm

**Input:** Given a point  $p$  and non-null quadtree  $Q$

**Output:** The label NW, NE, SW, SE of quadrant in which point  $p$  belongs with respect to the root of  $Q$

### Procedure:

- Step 1. if x-coordinate of  $p$  less than x-coordinate of root of  $Q$
- Step 2. then if y-coordinate of  $p$  less than y-coordinate of root of  $Q$
- Step 3. then return quadrant SW
- Step 4. else return quadrant NW
- Step 5. else if y-coordinate of  $p$  less than y-coordinate of root of  $Q$
- Step 6. then return quadrant SE
- Step 7. else return quadrant NE

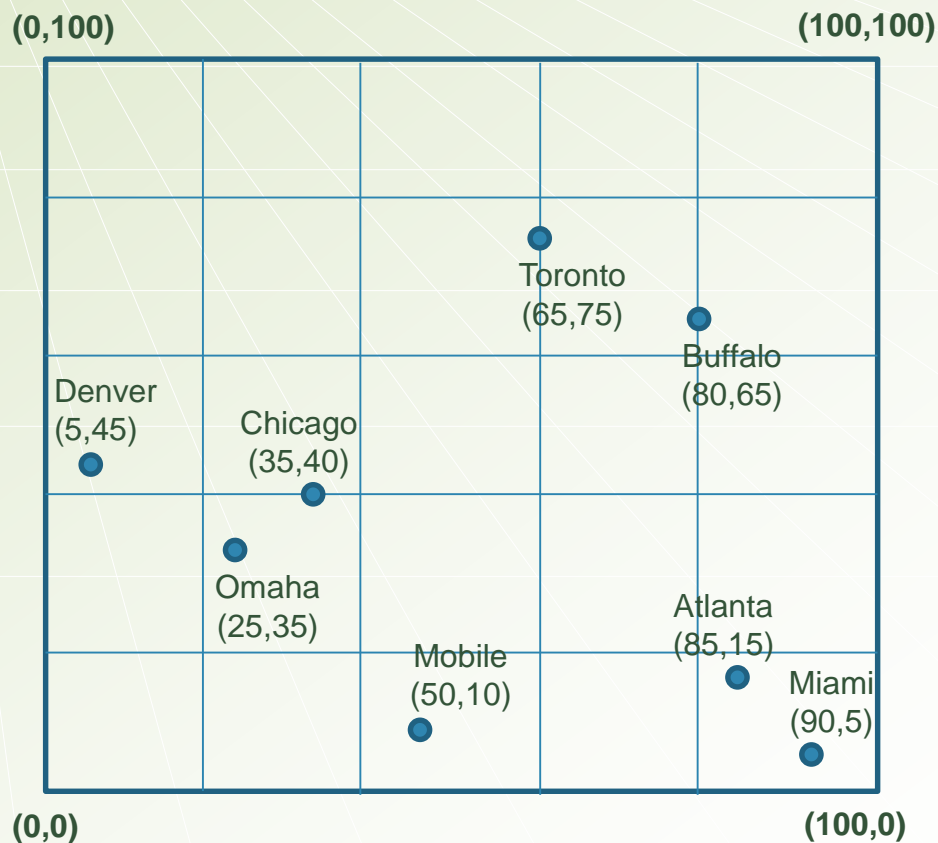


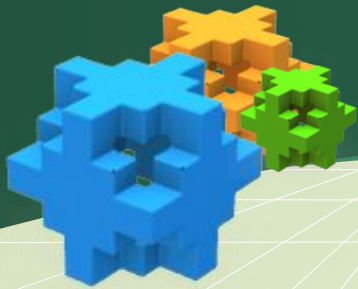


# Ví dụ Point Quadtree

❖ Xây dựng cây PQ với tập dữ liệu sau:

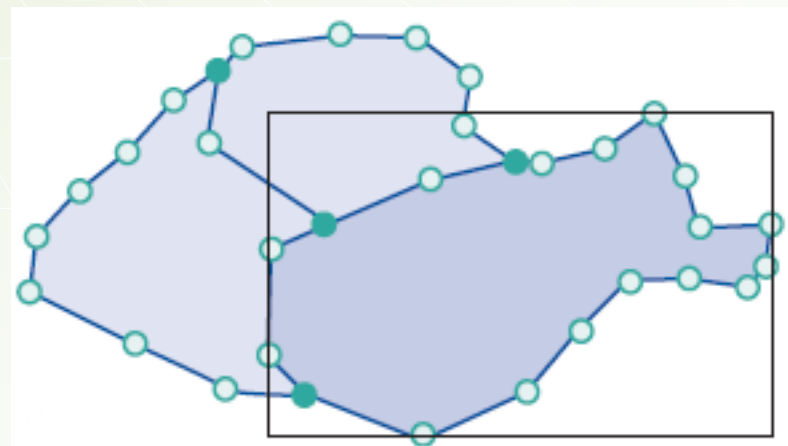
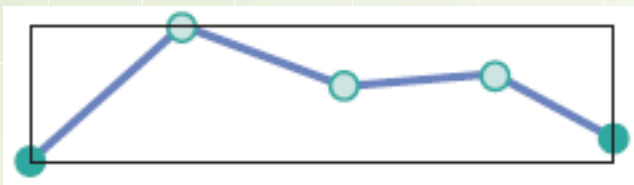
- Chicago (35, 40), Denver (5, 45), Toronto (60, 75), Omaha (25, 35), Mobile (50, 10), Buffalo (80, 65), Atlanta (85, 15), Miami (90, 5)

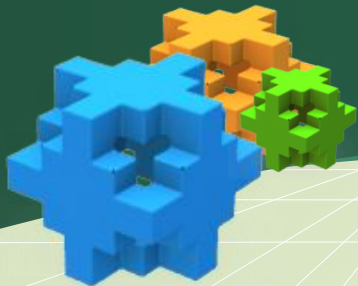




# R-tree

- ❖ R-tree nhóm các đối tượng điểm, đường và vùng trên cơ sở MBR (*Minimum Bounding Rectangle*) của chúng.
- ❖ Khái niệm MBR/MBB
  - Chữ nhật nhỏ nhất có cạnh song song với trục tọa độ, bao đặc trưng đường gấp khúc hay đa giác
  - Ví dụ ứng dụng: Kiểm tra hai đa giác cắt nhau? Trước hết kiểm tra MBR sau đó kiểm tra giao đa giác nếu cần.





# R-tree

- ❖ Mỗi node tương ứng với một trang đĩa từ
- ❖ Tổng số đầu vào của mỗi node (trừ root) trong cây trong khoảng  $m$  và  $M$ , với  $m \in [0, M/2]$ 
  - $M$ - Giá trị cực đại của đầu vào của một node
  - $M = \text{size}(P) / \text{size}(E)$ ;  
     $P$ - disk page và  
     $E$ - entry
- ❖ Các node lá có cùng mức
- ❖ Nút gốc có ít nhất hai đầu vào (*children*)
- ❖ Khuôn mẫu của nodes
  - $\{(MBR; \text{obj\_ptr})\}$  for leaf nodes
  - $\{(MBR; \text{node\_ptr})\}$  for non-leaf nodes

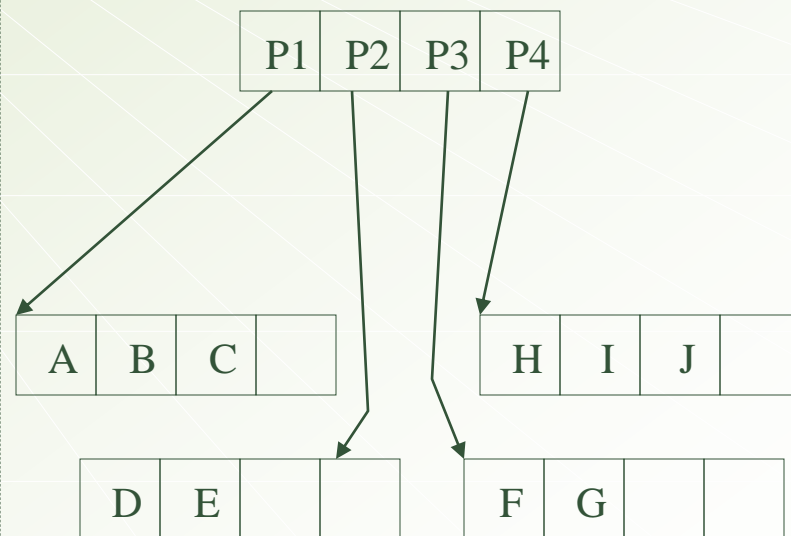
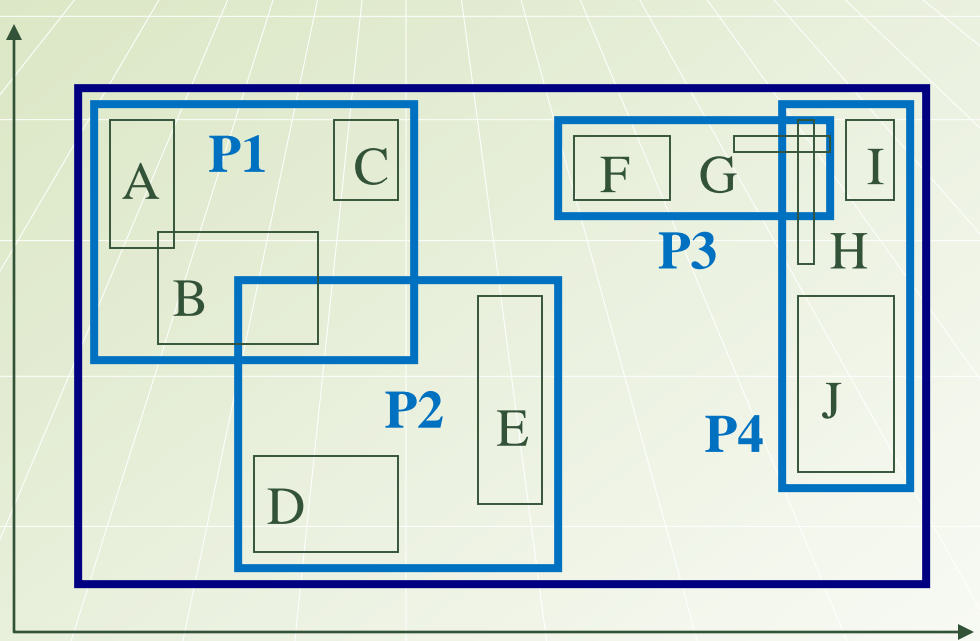


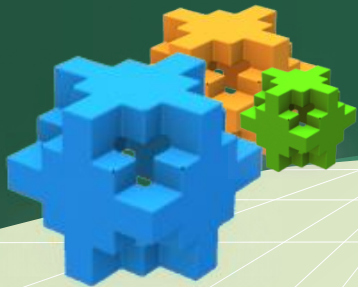


# R-tree

## ❖ Ví dụ R-tree

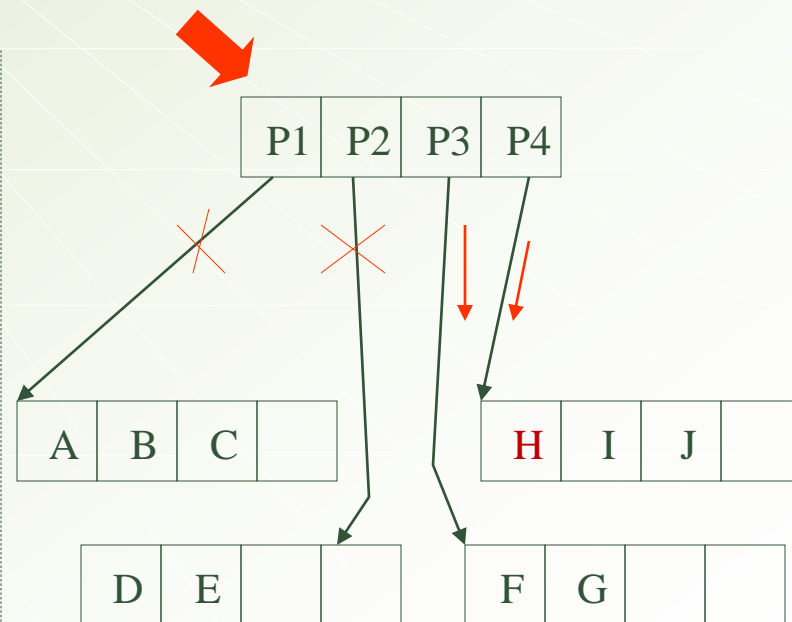
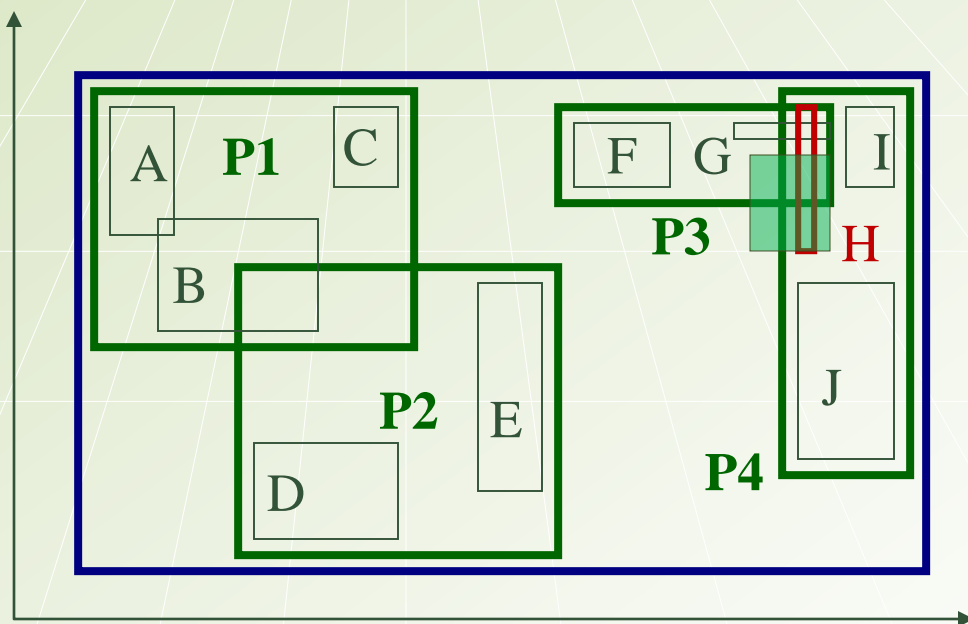
- Nhóm các chữ nhật gần nhau vào một MBR cha
- Ví dụ với  $M=4$  và  $m=2$ :

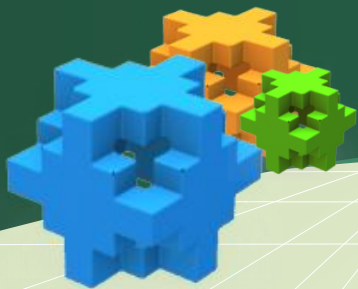




# R-trees: Search

- ❖ Mỗi node cha phủ hoàn toàn các nodes con
- ❖ MBR của con có thể bao phủ bởi nhiều cha, nhưng chỉ lưu trữ dưới một cha.
- ❖ Ví dụ với Window query W

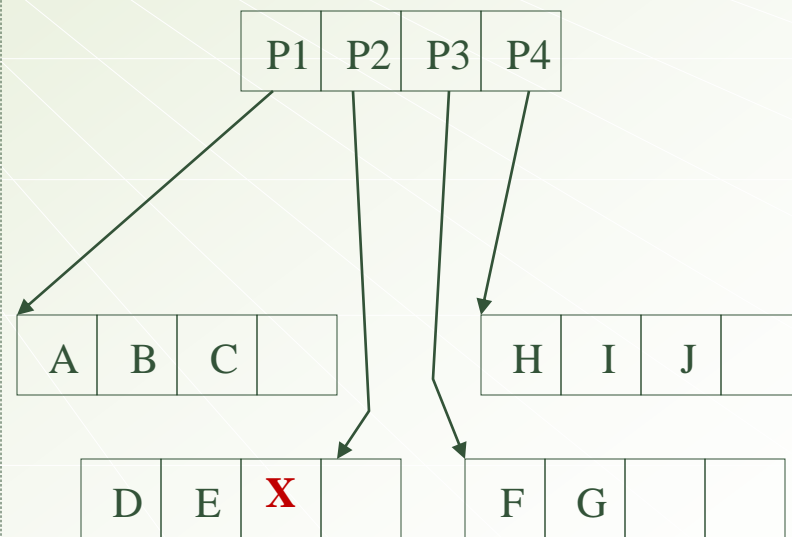
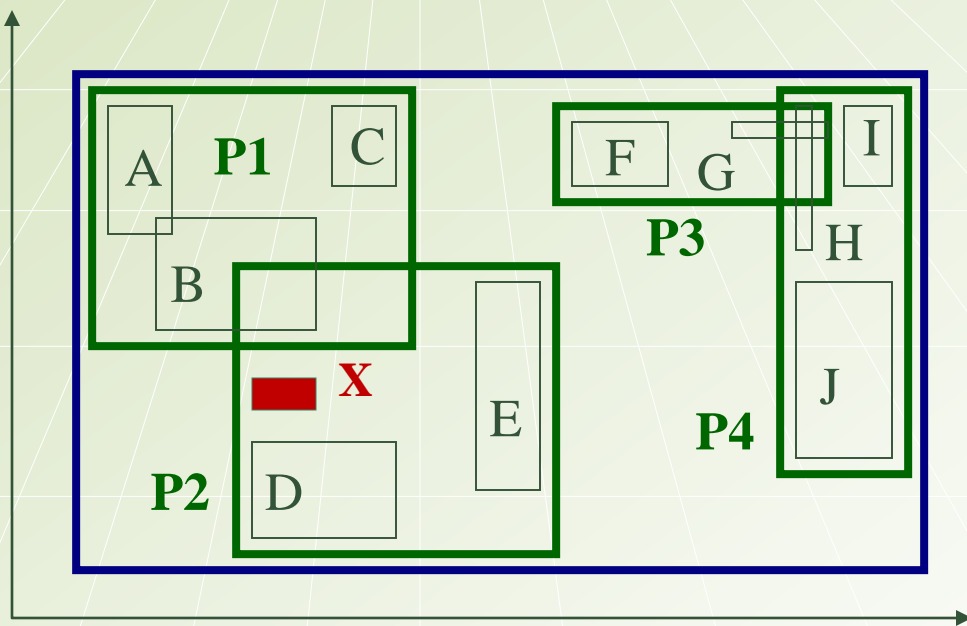


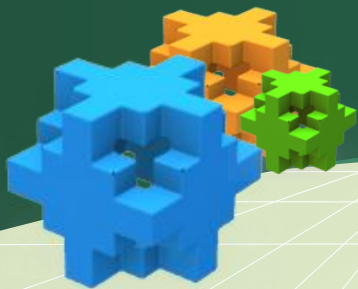


# R-trees: Insertion

## ❖ Chèn **X**

- Duyệt Top-Down từ gốc cây R

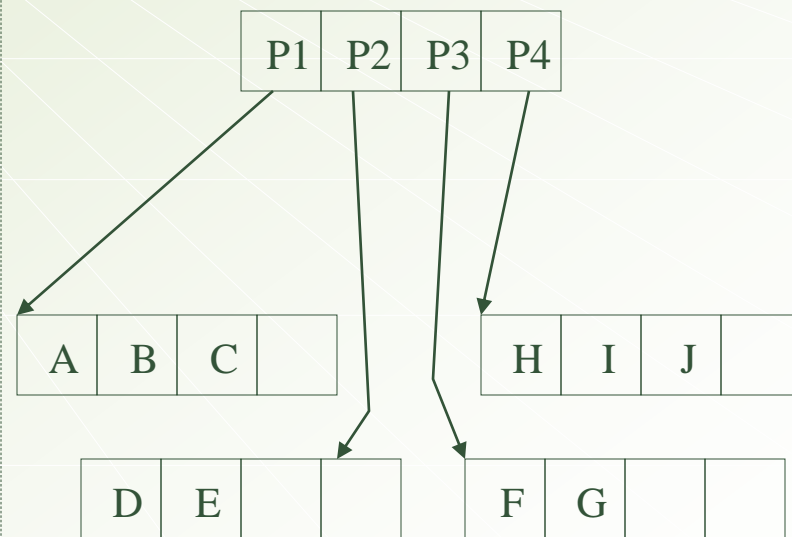
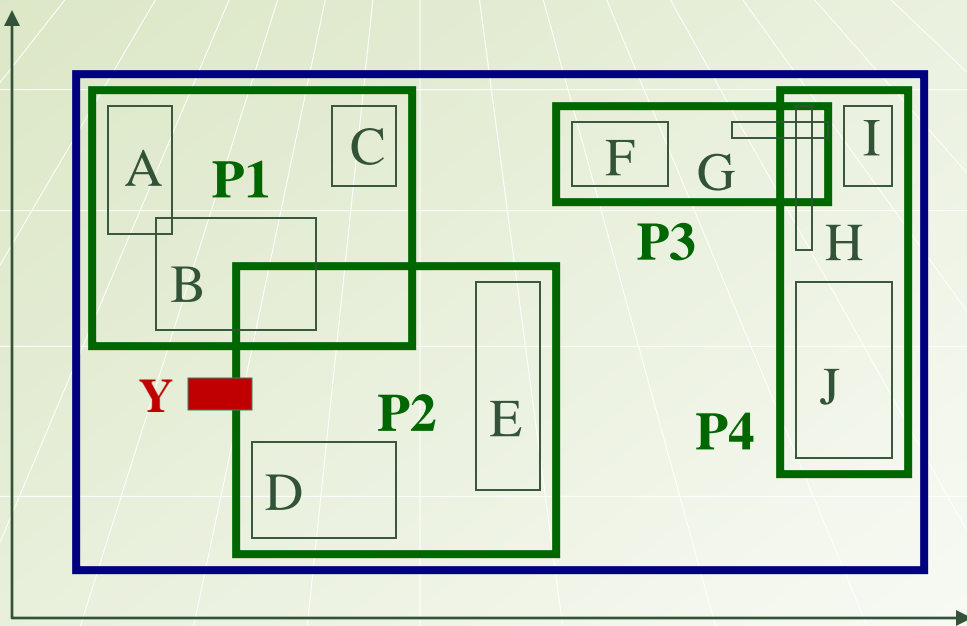




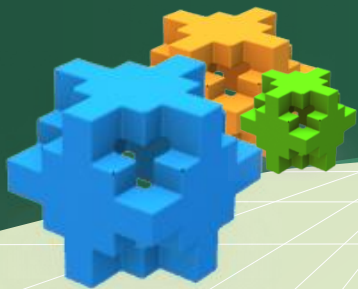
# R-trees: Insertion

## ❖ Chèn **Y**

- Tìm chữ nhật chứa Y để mở rộng



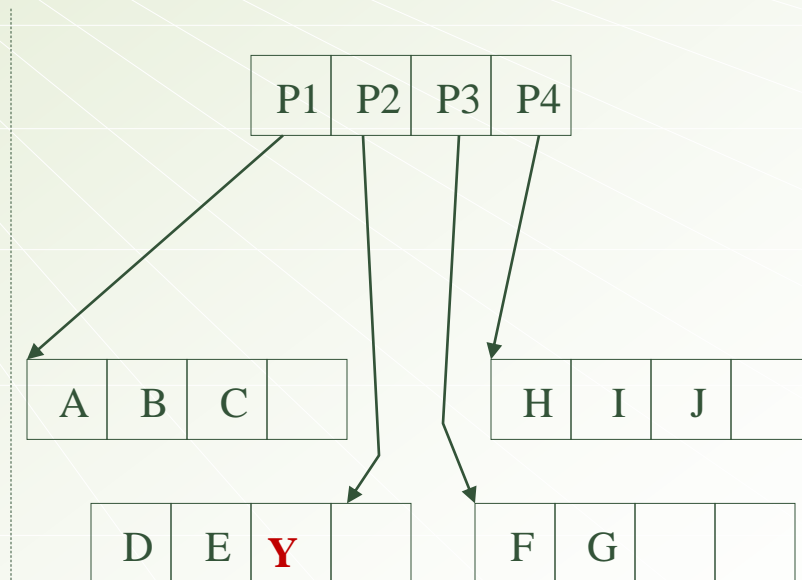
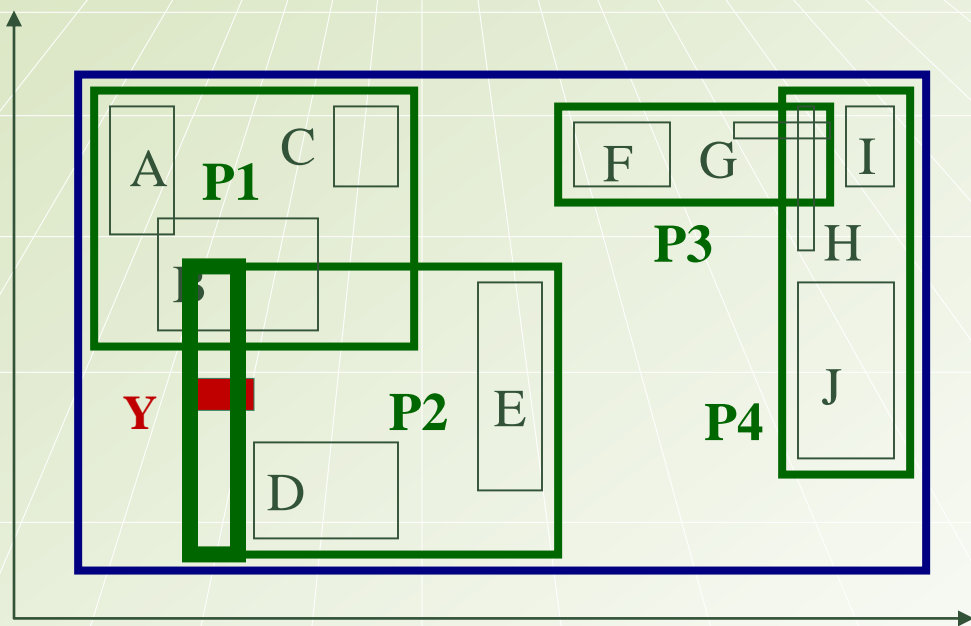




# R-trees: Insertion

## ❖ Mở rộng MRB của cha

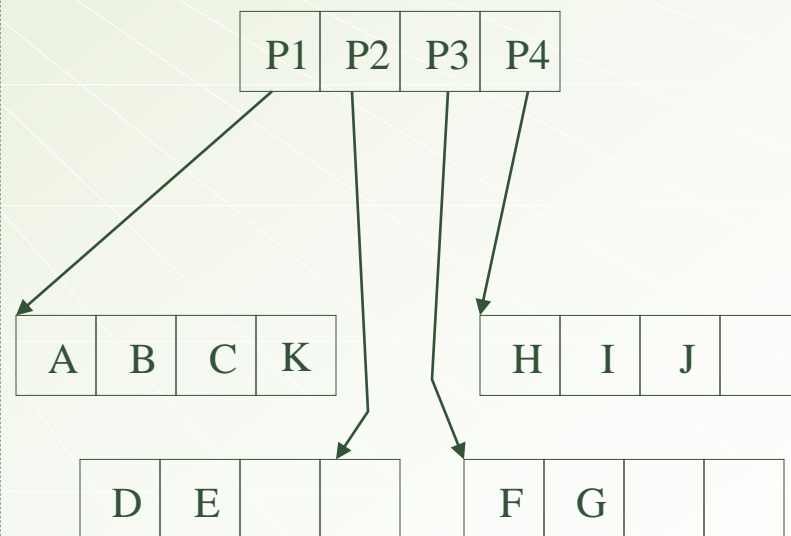
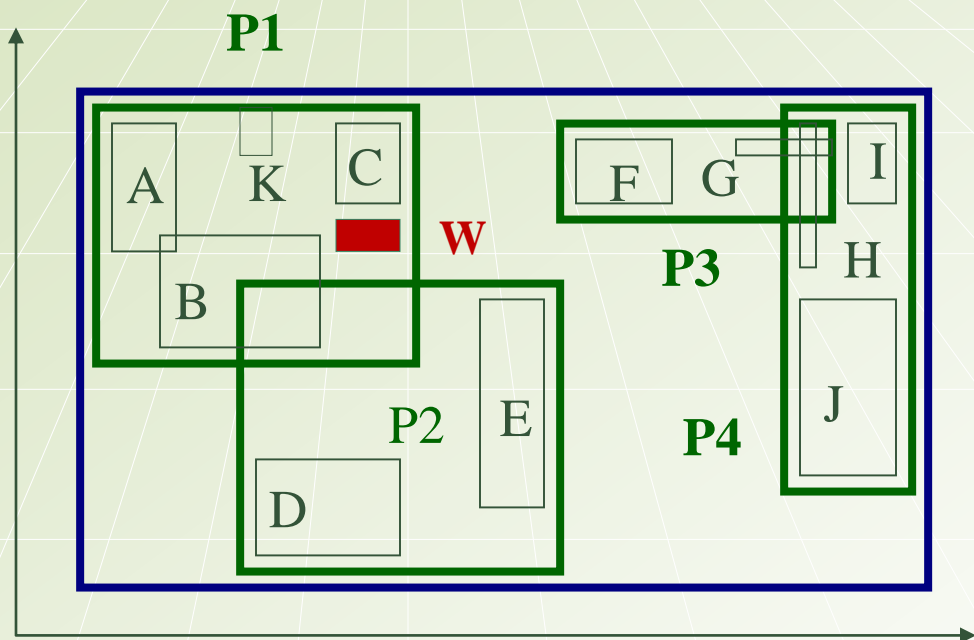
- Tìm entry nào mà chỉ cần ít mở rộng để có thể chứa **Y** (Ràng buộc diện tích nhỏ nhất)

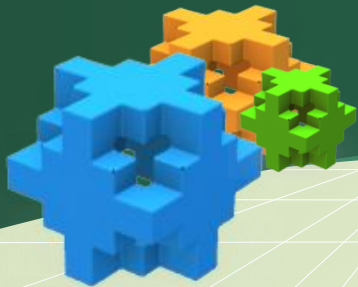




# R-trees: Insertion

- ❖ Tìm node để chèn **W**
  - Nếu node đầy?

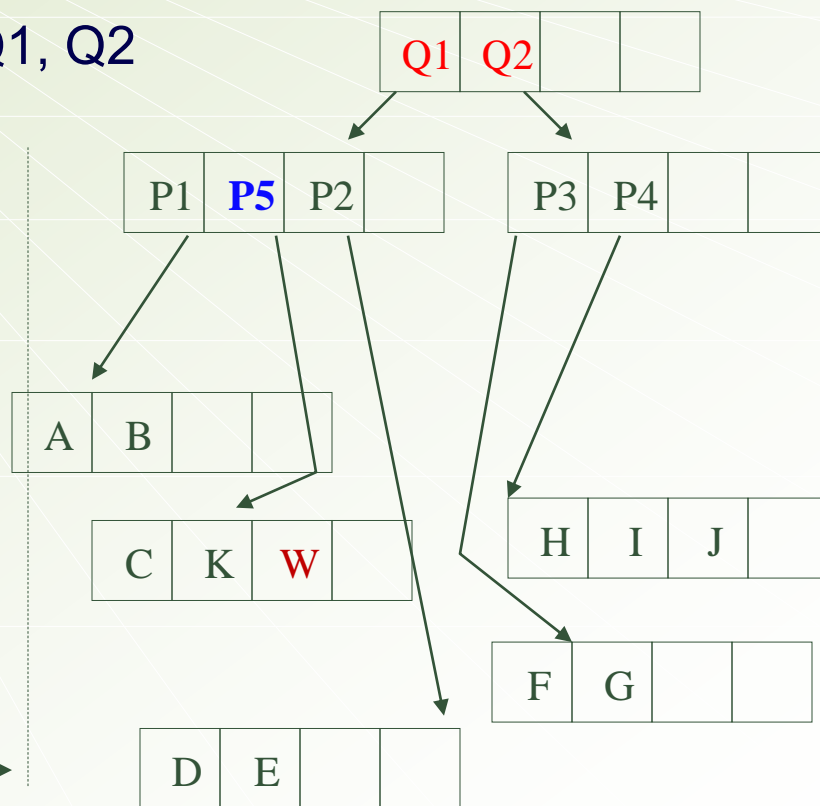
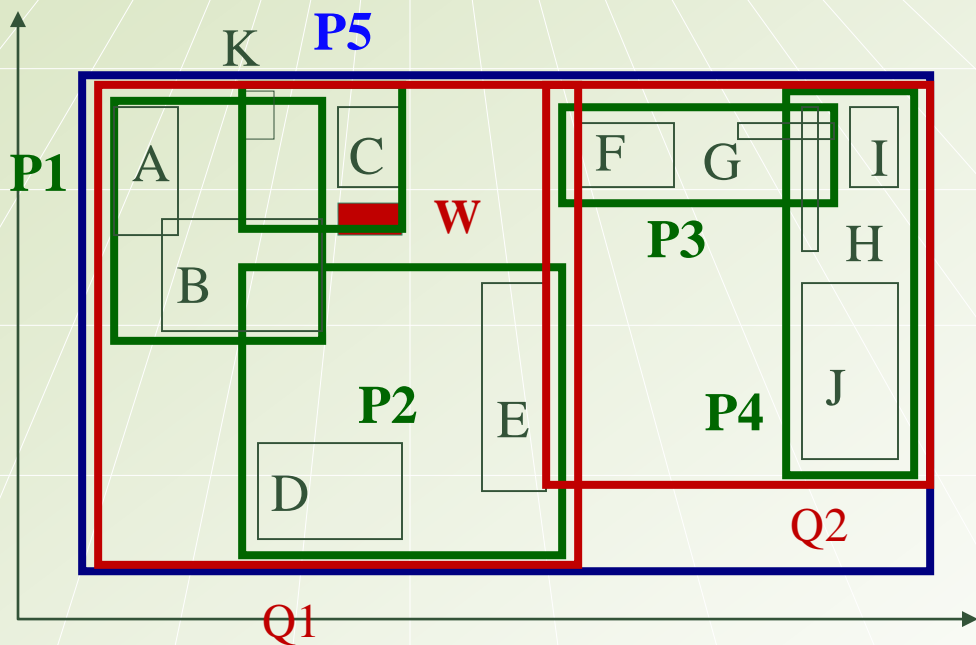


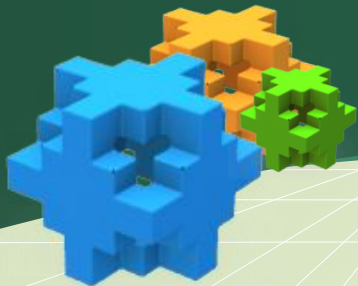


# R-trees: Insertion

## ❖ Tìm node để chèn **W**

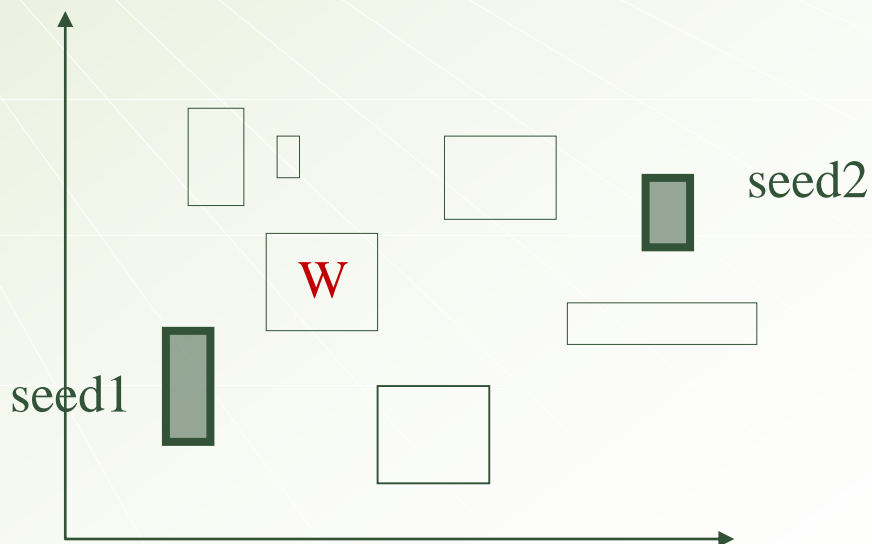
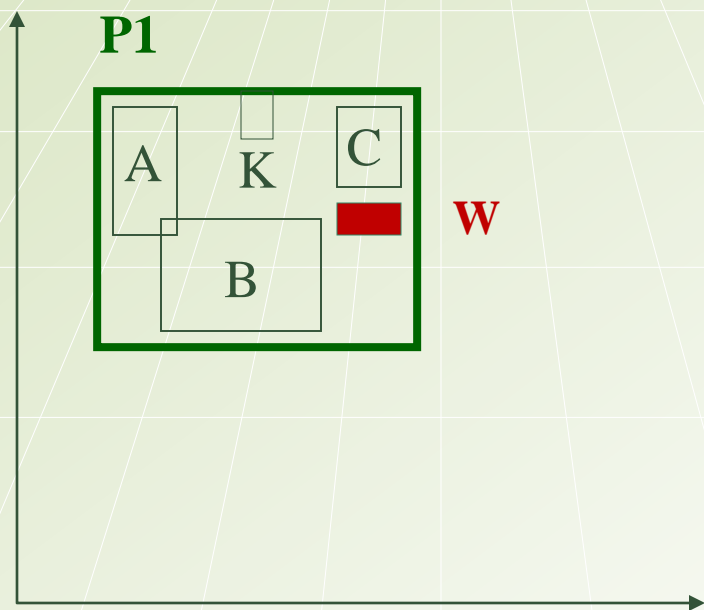
- Nếu node đầy thì phải bẻ đôi: P1 thành P1, P5 và node gốc bẻ đôi thành Q1, Q2



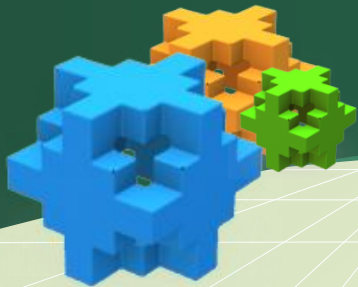


# R-trees: Split

- ❖ Chiến lược bề node: Phân chia các MRB thành hai nhóm
  - Chọn hai chữ nhật làm “seeds”
  - Gán chữ nhật **W** vào “seed” gần hơn

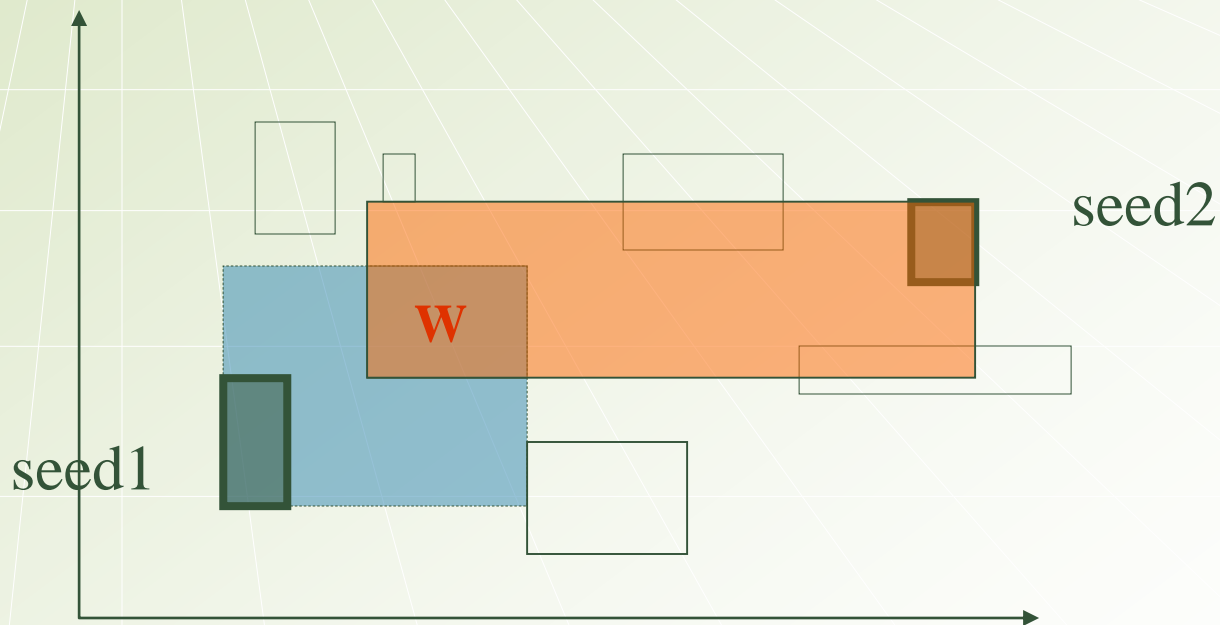






# R-trees: Split

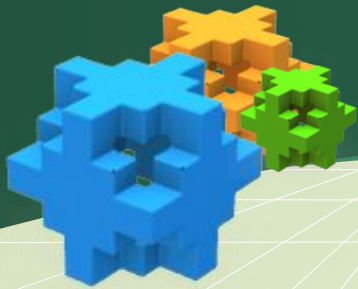
- ❖ Khái niệm “gần hơn”:
  - Ít tăng diện tích hơn





# R-tree

- ❖ R-Tree là phương pháp chỉ mục dữ liệu địa lý thông dụng vì sự mềm dẻo và hiệu năng rất cao của chúng
  - Phù hợp hơn với dữ liệu đa chiều
  - Truy vấn nhanh, cập nhật chậm hơn Quatree và lưới
  - Sử dụng trong Informix Spatial Datablade và Oracle Spatial



## 4. Tổng kết bài

- ❖ Đã nghiên cứu các kiểu hiện tượng địa lý khác nhau
- ❖ Khảo sát cách thức trình diễn trong máy tính bằng GIS
- ❖ Phân biệt các hiện tượng:
  - Trường (liên tục, rời rạc),
  - Đối tượng
- ❖ Cấu trúc cây dữ liệu

**Câu hỏi?**