



Đại học quốc gia thành phố Hồ Chí Minh  
Trường đại học Công nghệ thông tin

# K-Dimensional Tree

- 19521322 - Huỳnh Ngọc Công Danh
- 20520519 - Phạm Xuân Hoàng
- Giảng viên hướng dẫn: Nguyễn Thanh Sơn



# Nội dung



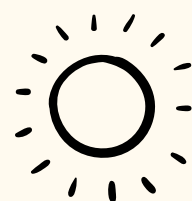
**1. Giới thiệu**



**2. Các thao tác cơ bản**



**3. Ứng dụng**



**4. Ưu và nhược điểm**



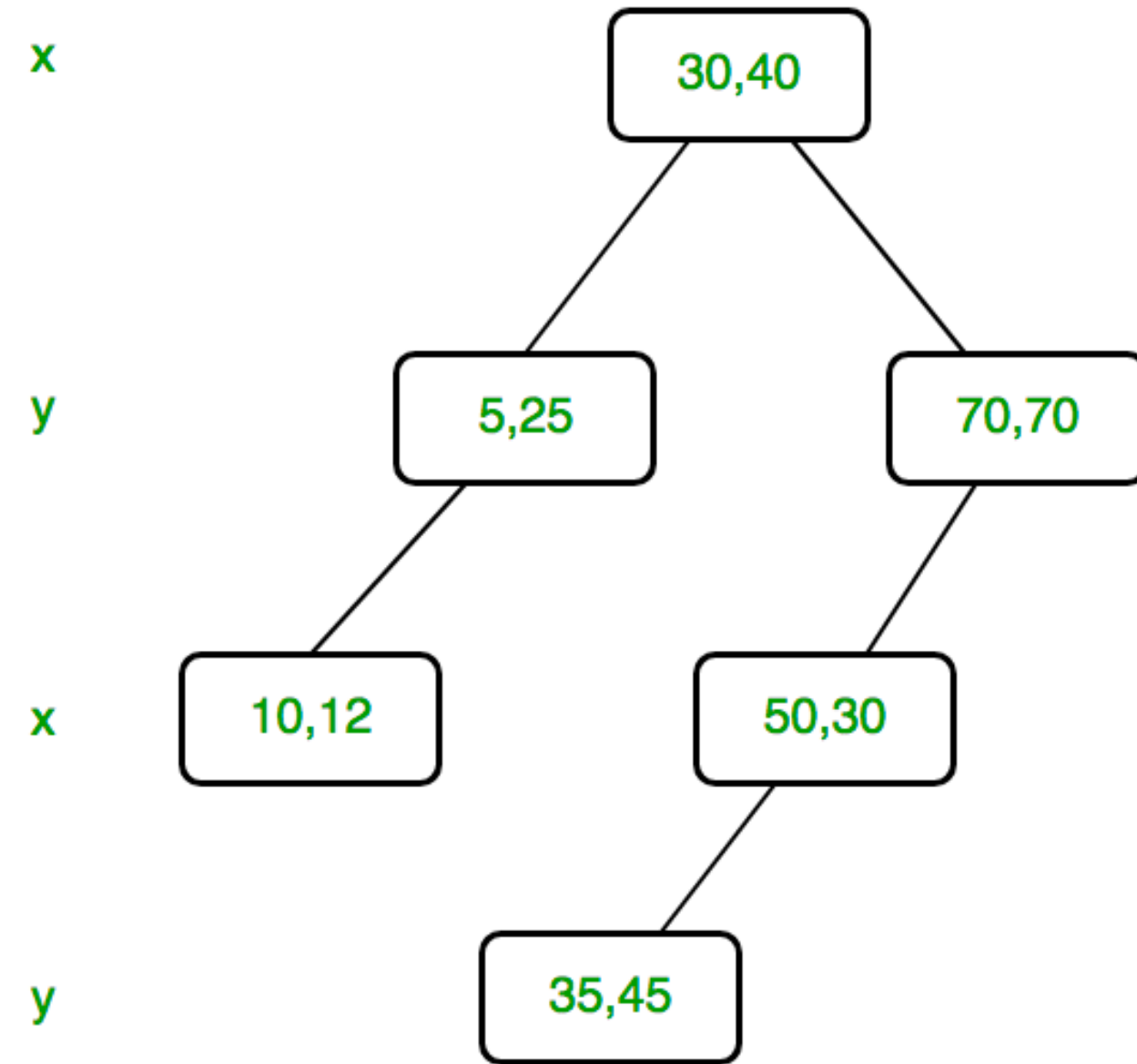
**5. Demo**

# 1. Giới thiệu

**Bài Toán :** Giả sử bạn là một người mới chuyển đến sống và bạn muốn biết xem cửa hàng tiện lợi trong khu vực gần nhà của bạn.  
Thông thường thì nếu không sử dụng google map thì bạn sẽ phải đến từng cửa hàng để xem cái nào gần nhất.

# K-Dimensional Tree:

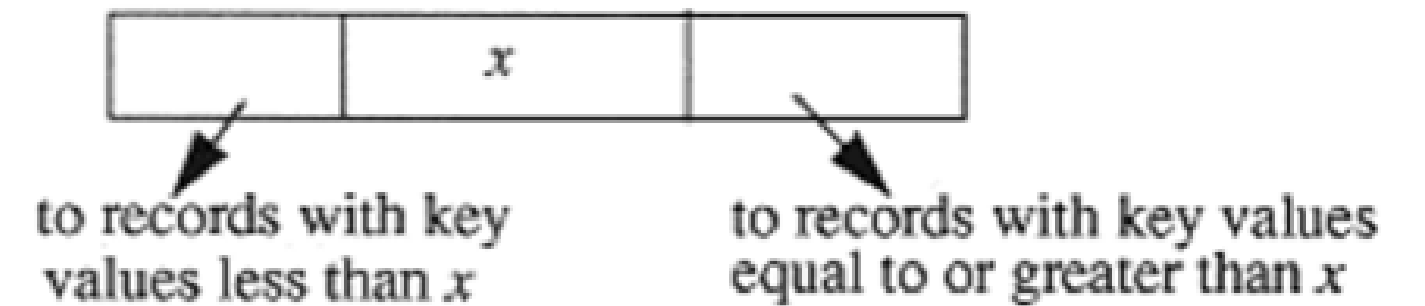
- K-Dimensional Tree thuộc họ Nearest neighbor (NN) search.
- K-Dimensional Tree khác với thuật toán cây nhị phân khác thay vì mỗi node trong cây nhị phân có một giá trị thì mỗi node có K giá trị tương đương với K chiều trong không gian tọa độ



# Cấu tạo 1 node

- Mỗi node gồm vector chứa  $k$  khóa
- Ở mỗi mức, các khóa sẽ được chia theo giá trị của 1 chiều nhất định

- Mức 0: giá trị chiều 0
- Mức 1: giá trị chiều 1, ...
- Mức  $k-1$ : giá trị chiều  $k-1$
- Mức  $k$ : giá trị chiều 0, ...



## 2. Các thao tác cơ bản



**Thêm**

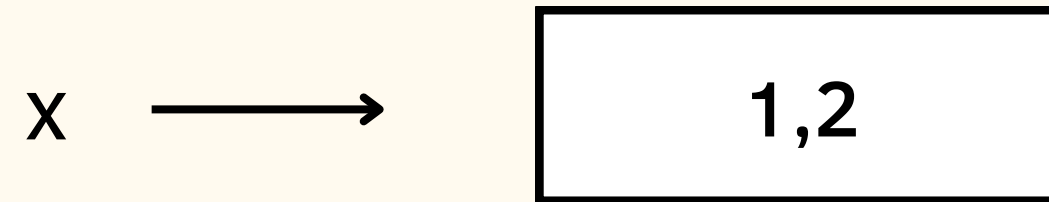
**Tìm kiếm**

**Xóa**

# **Thêm trong K-Dimensional Tree**

# Thêm Node

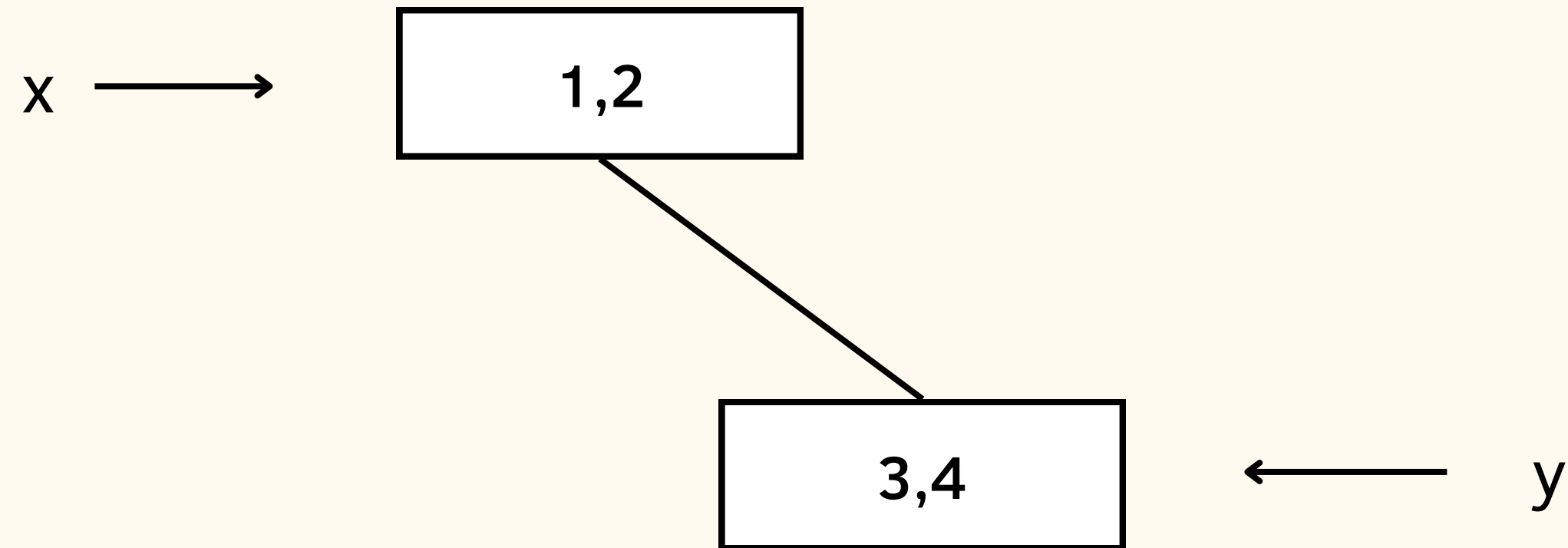
Thêm node (1,2)





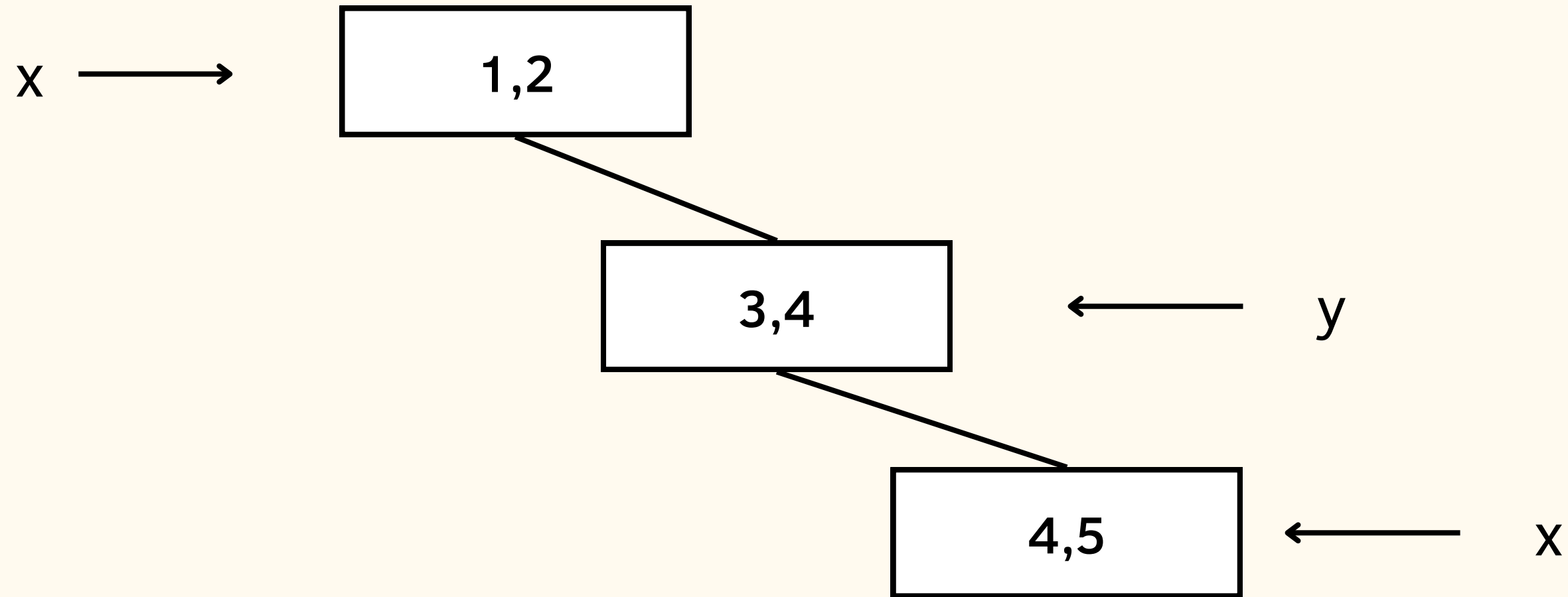
# Thêm Node

Thêm node (3,4)



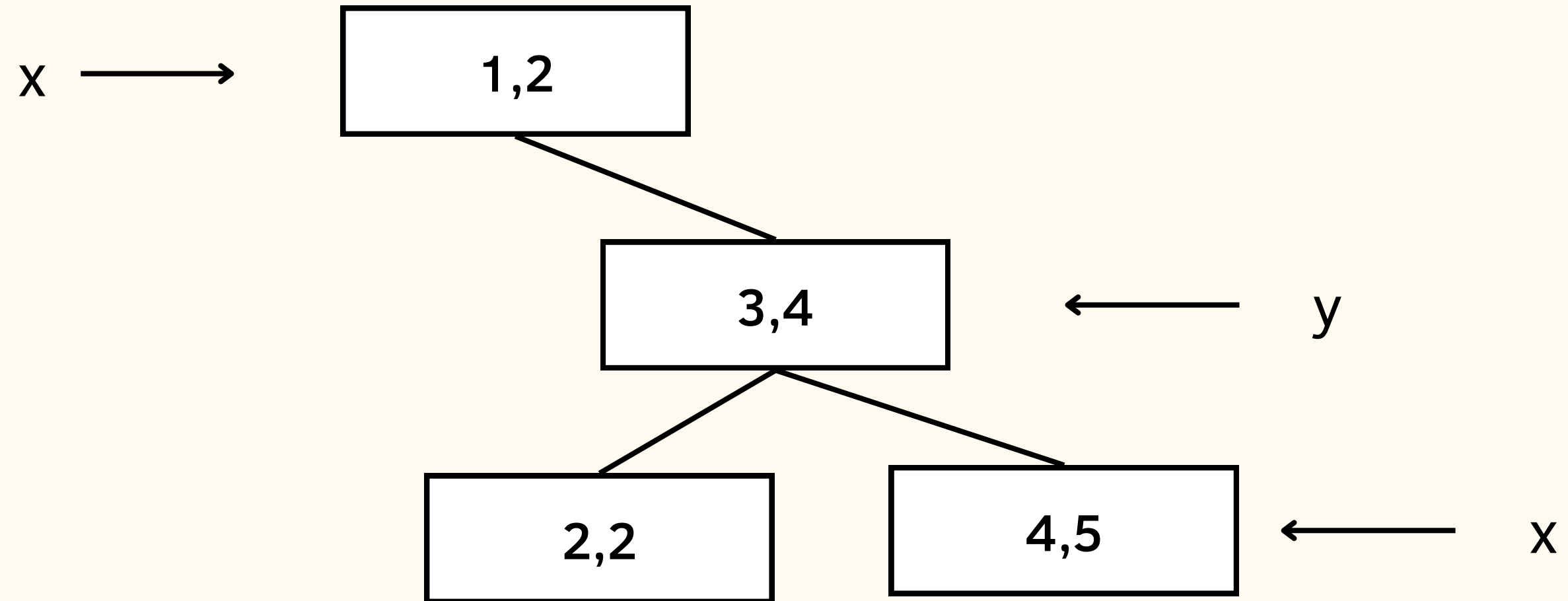
# Thêm Node

Thêm node (4,5)



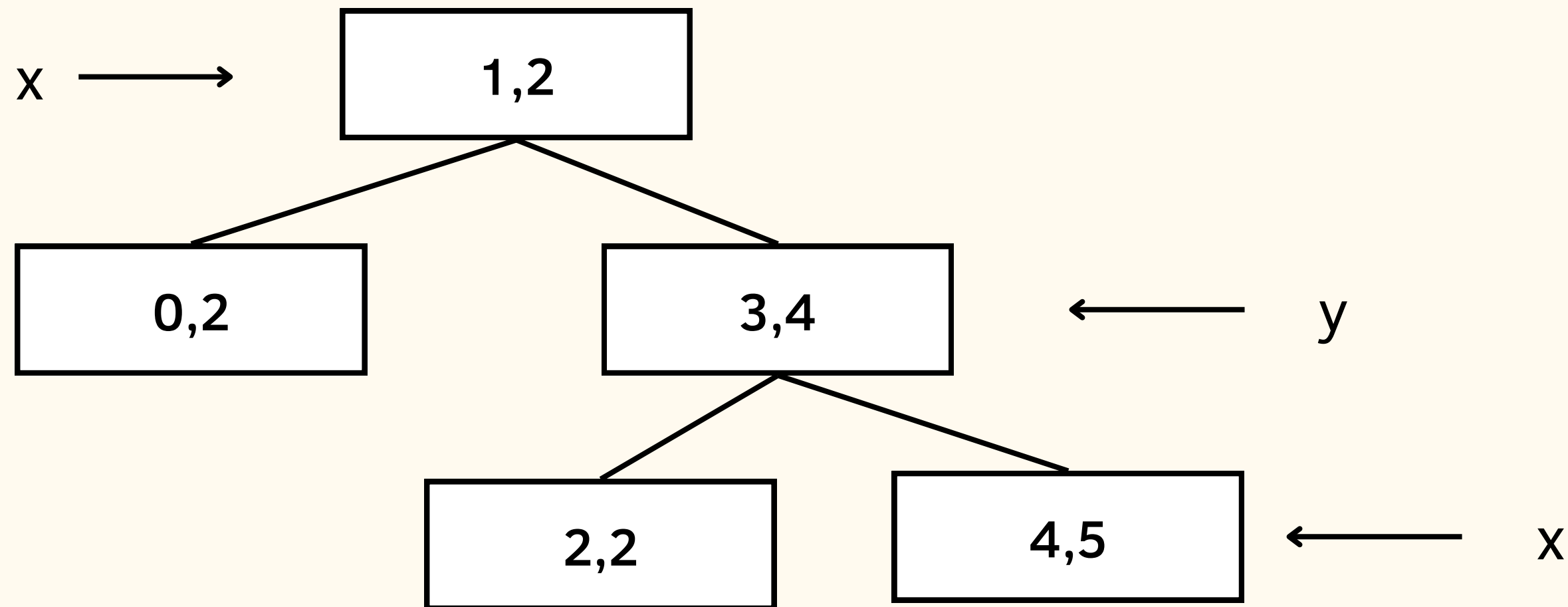
# Thêm Node

Thêm node (2,2)



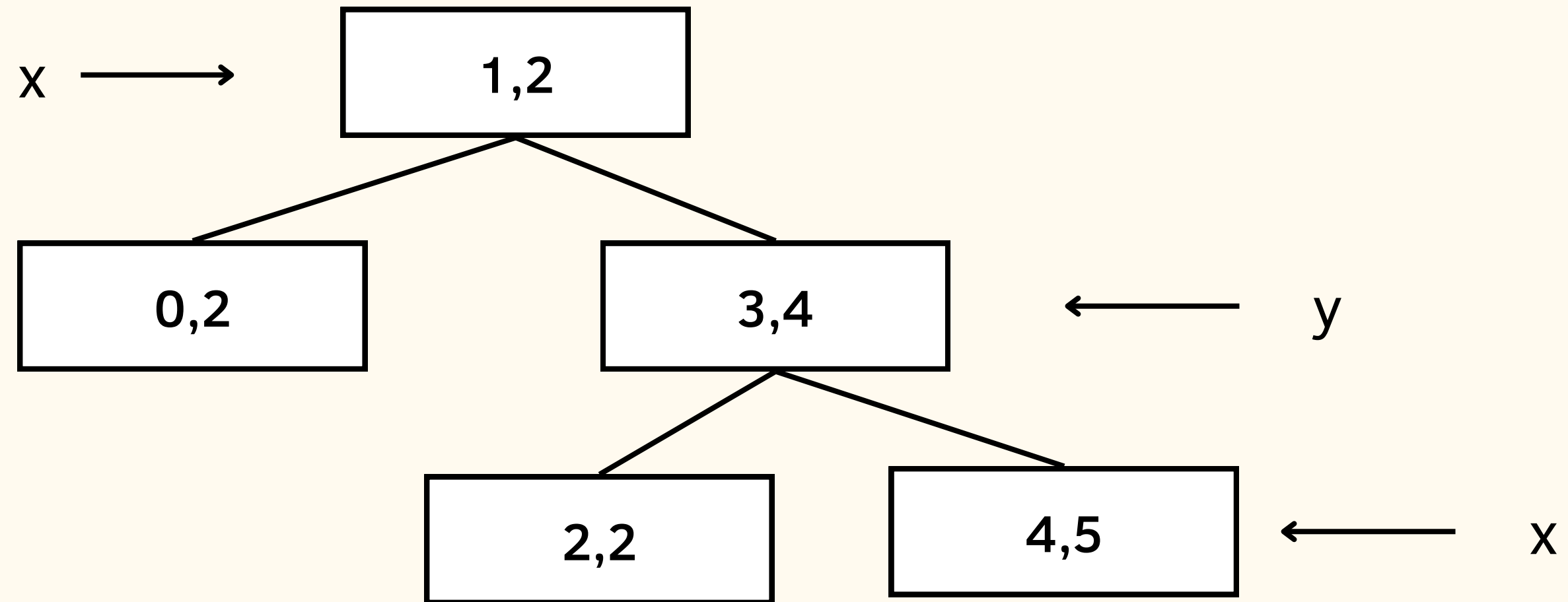
# Thêm Node

Thêm node (0,2)



# Thêm Node

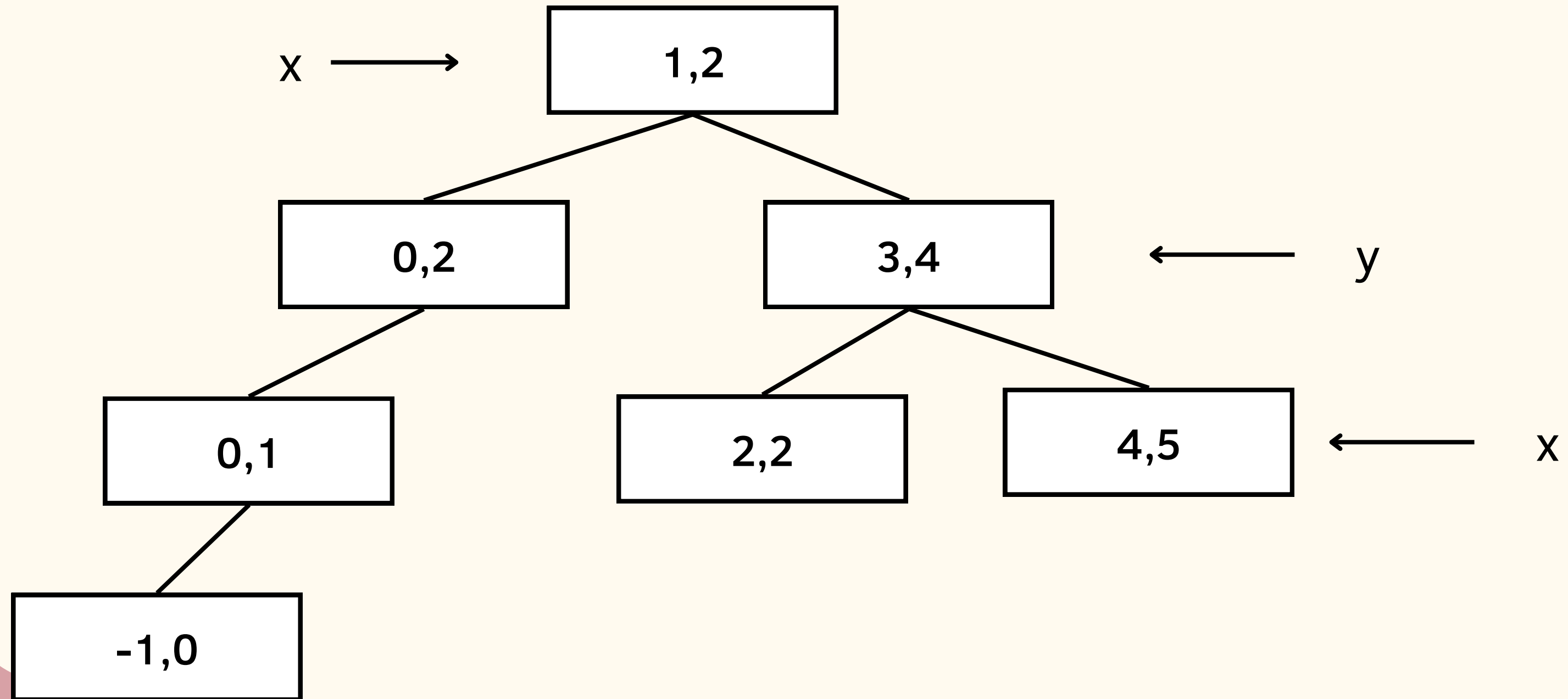
Thêm node (0,2)



# **Tìm kiếm trong K-Dimensional Tree**

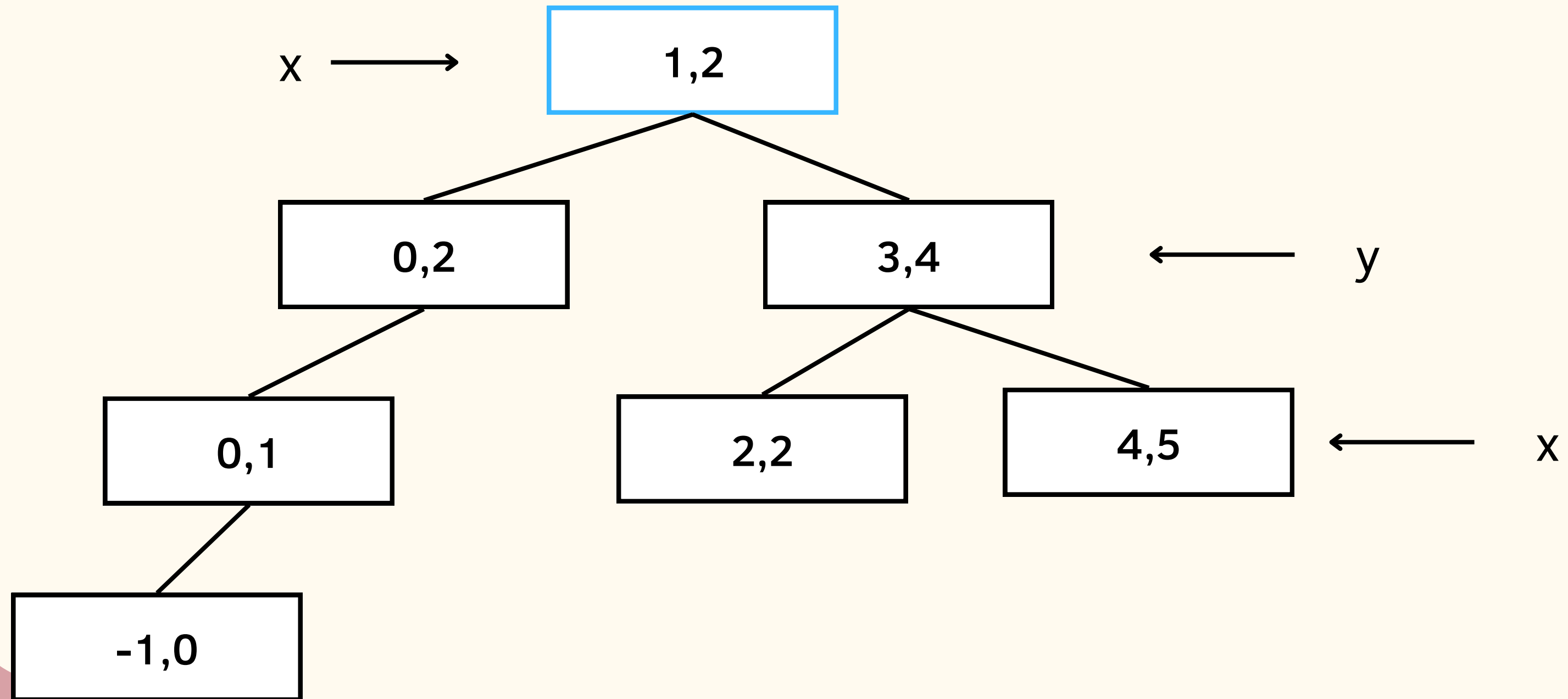
# Tìm kiếm Node

Tìm kiếm node (2,2)



# Tìm kiếm Node

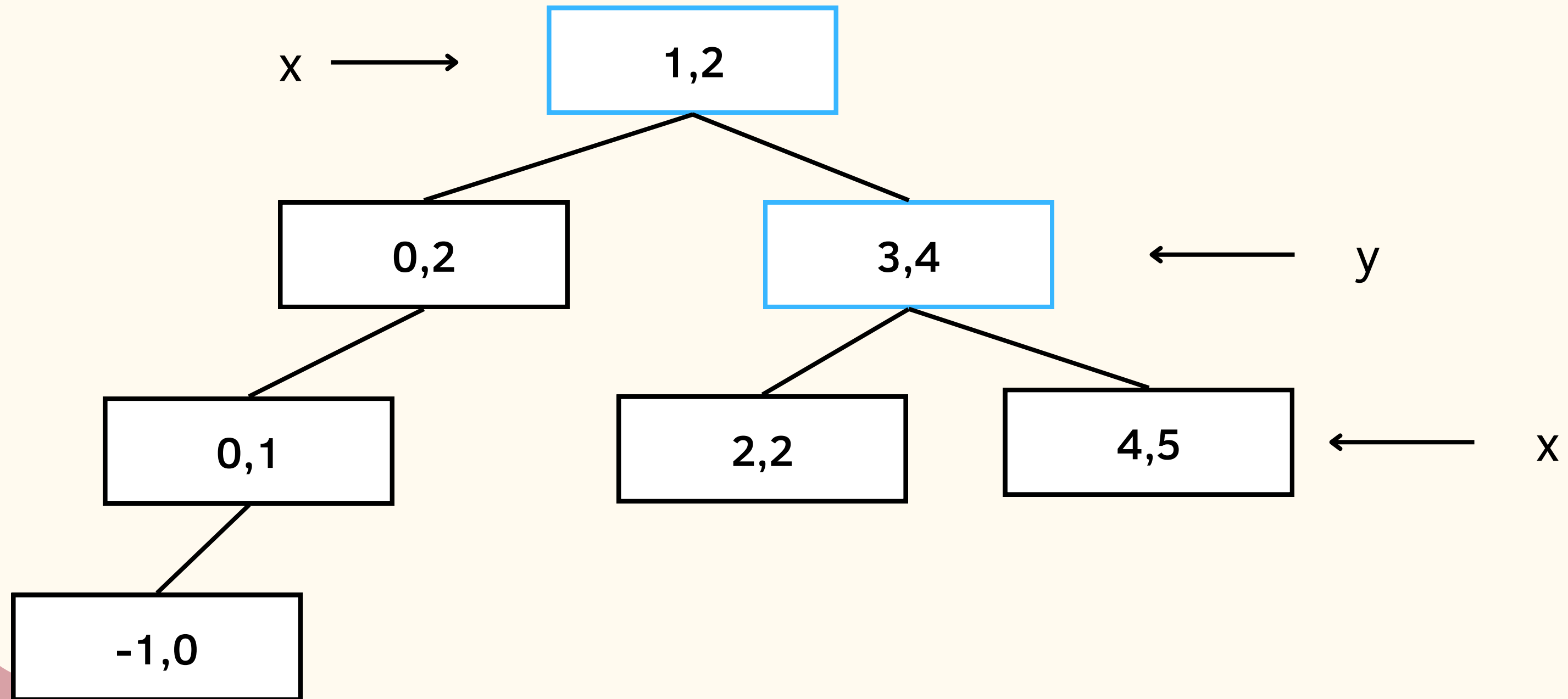
Tìm kiếm node (2,2)





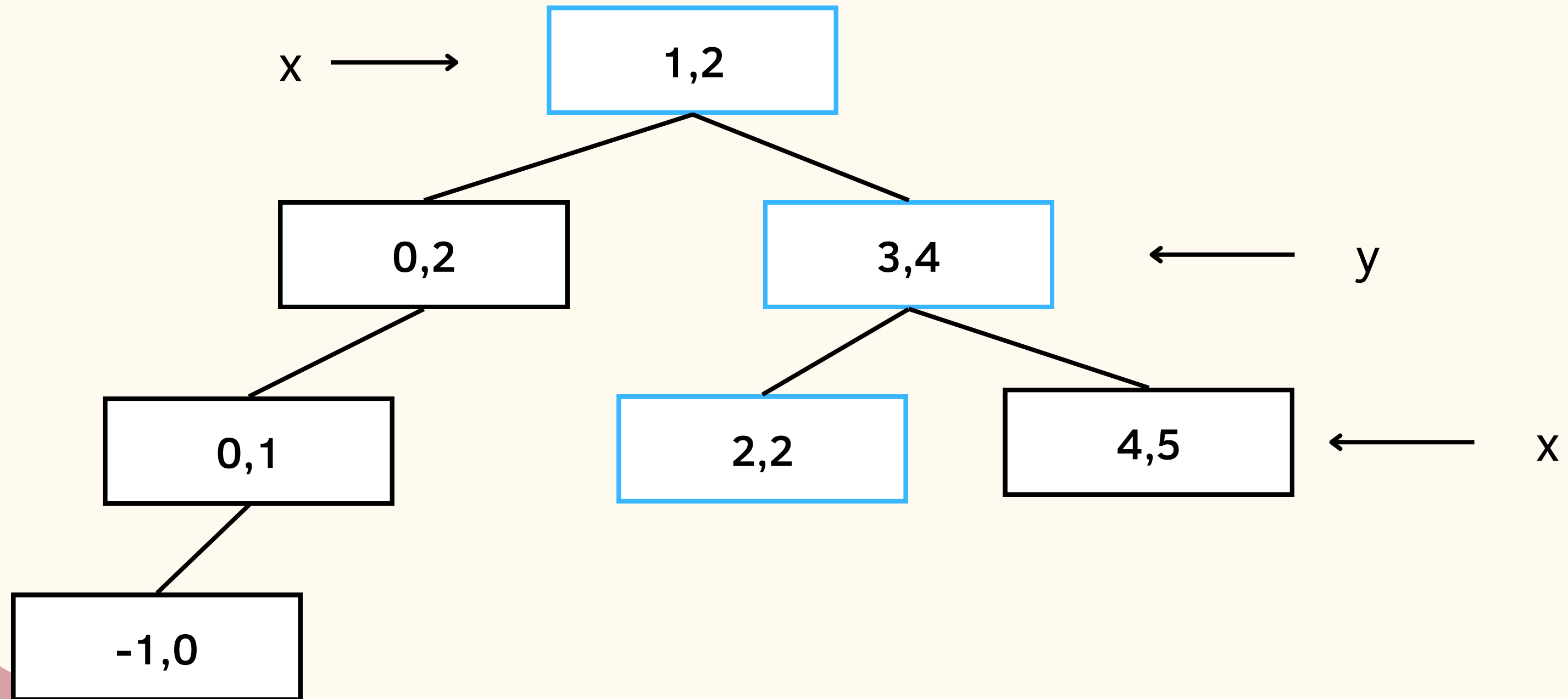
# Tìm kiếm Node

Tìm kiếm node (2,2)



# Tìm kiếm Node

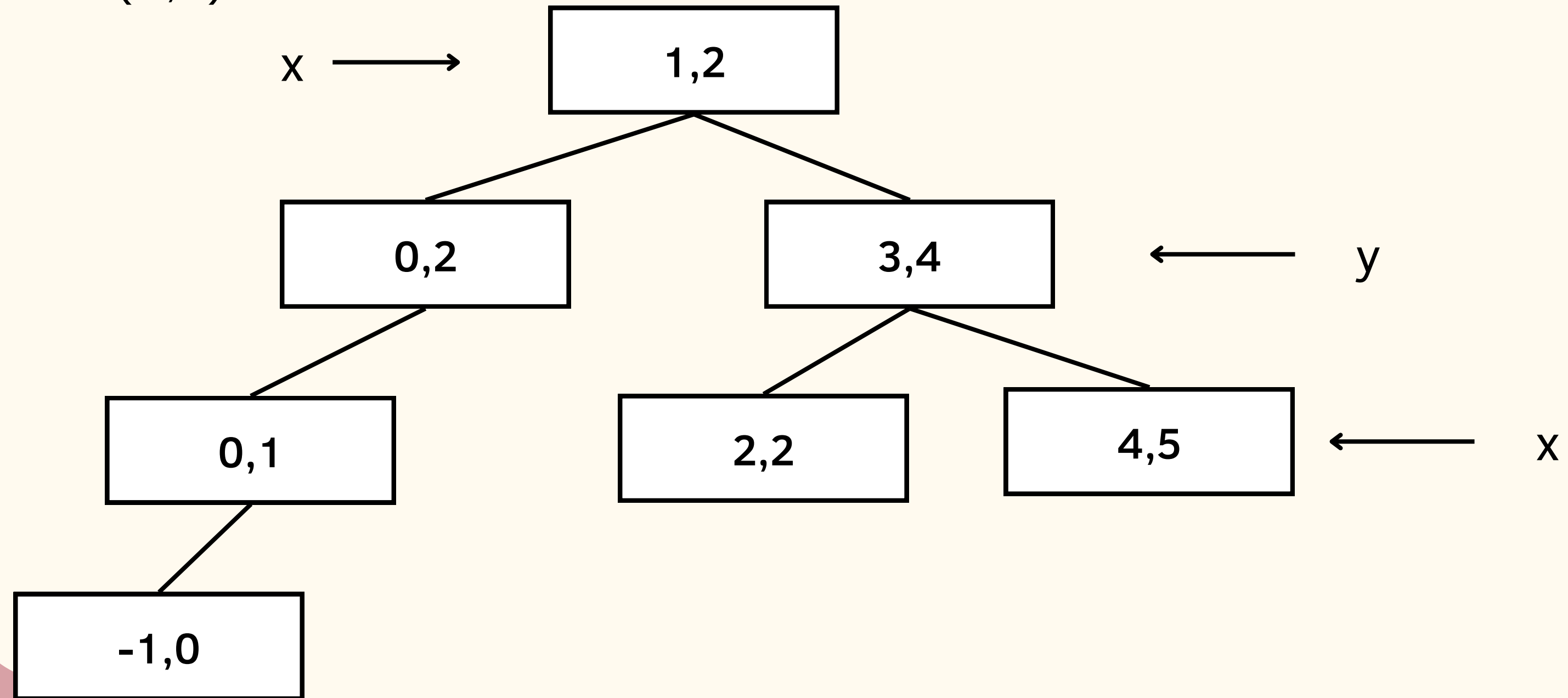
Tìm kiếm node (2,2)



# **Xoá trong K-Dimensional Tree**

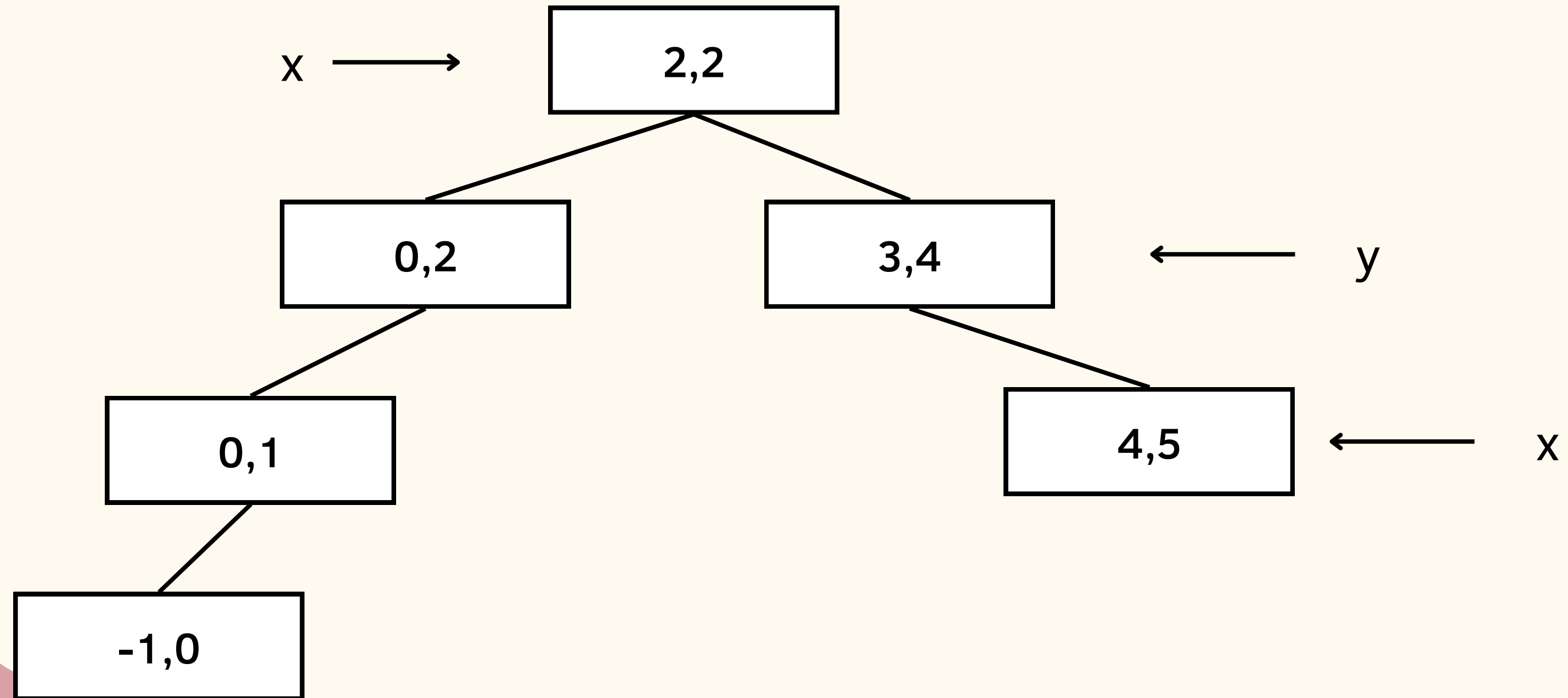
# Xóa Node

Xóa node (1,2)



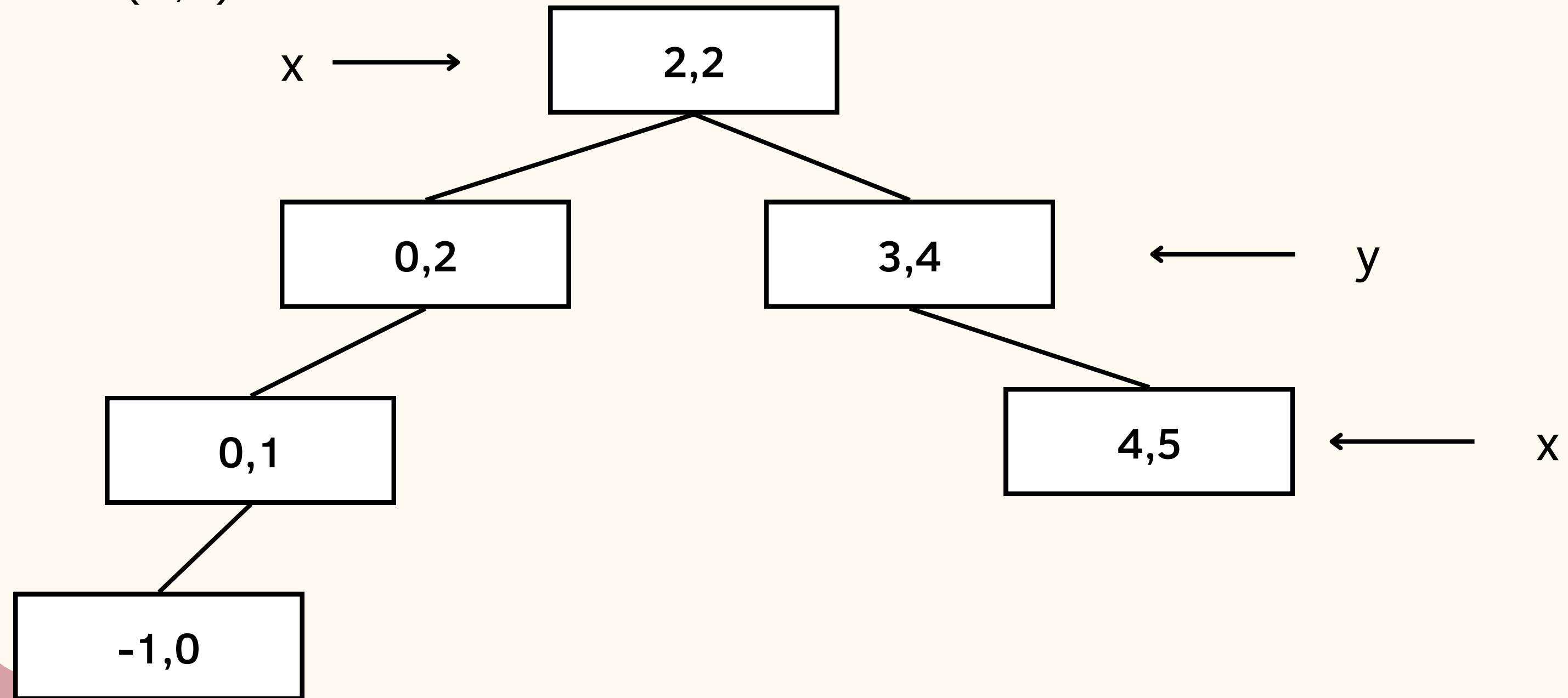
# Xóa Node

Sau khi xóa node (1,2)



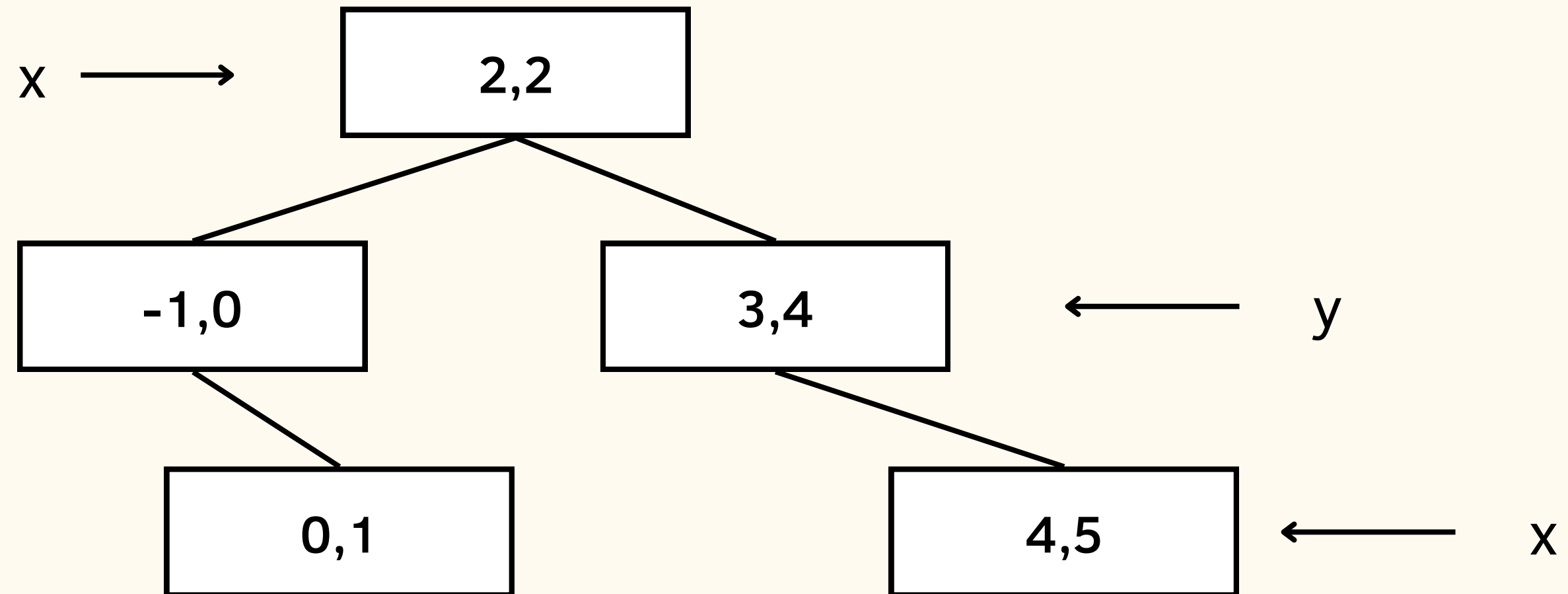
# Xóa Node

Xóa node (0,2)



# Xóa Node

Sau khi xóa node (0,2)



A large, abstract, pinkish-red shape on the left side of the slide, resembling a stylized letter 'C' or a partial circle.

## **3. Ứng dụng**



# KNN

Với số điểm  $n$ , số chiều  $d$ , và số điểm  $k$

Sử dụng kNN -> tìm điểm nhỏ nhất qua từng chiều -> quá phức tạp với dữ liệu nhiều chiều -> Một lần  
Time Complexity =  $O(knd)$

Dùng KD Tree -> Giảm được số chiều phải xử lý để tìm kiếm Nearest Neighbor -> Time Complexity =  
 $O(k \cdot \log(n))$

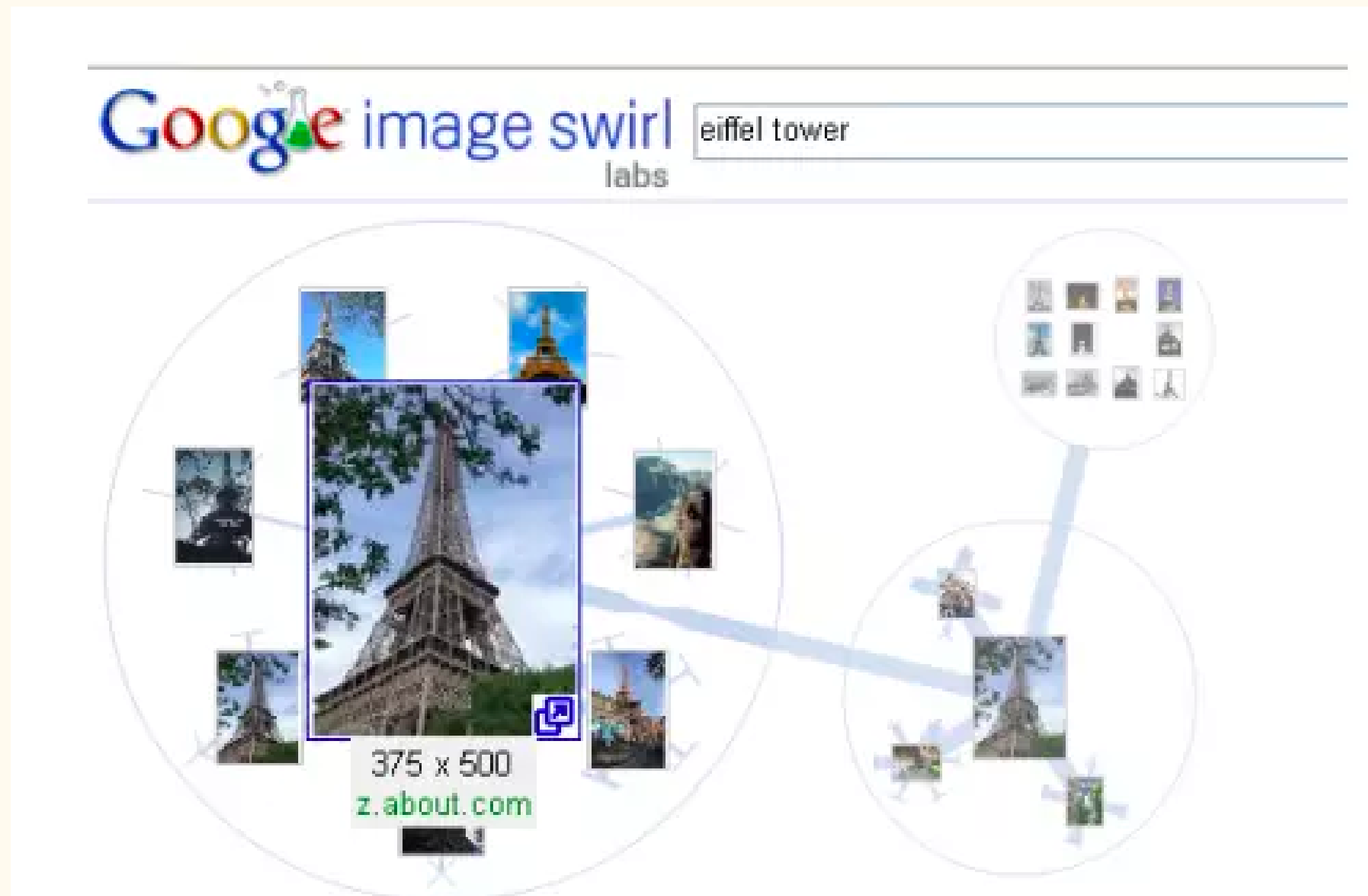
# Ray Tracing



Ray tracing là một kỹ thuật kết xuất để tạo ảnh bằng cách dò đường đi của ánh sáng dưới dạng pixel trong mặt phẳng ảnh và mô phỏng các hiệu ứng khi nó gặp các vật thể ảo.

Khi chơi các game 3d, các pixel 3d được xây dựng trong KD Tree -> chỉ cần xử lý các pixel lân cận, trong phạm vi cần thiết để tạo hiệu ứng như đổ bóng, phản chiếu, phát xạ,....,

# Sử dụng trong các bài toán với ảnh ảnh



# Optimization

Sử dụng KD Tree để cải tiến, ứng dụng vào các bài toán tìm kiếm khác như BKD Tree ( tìm kiếm ảnh ) , CKD Tree (phân lớp ảnh), ...



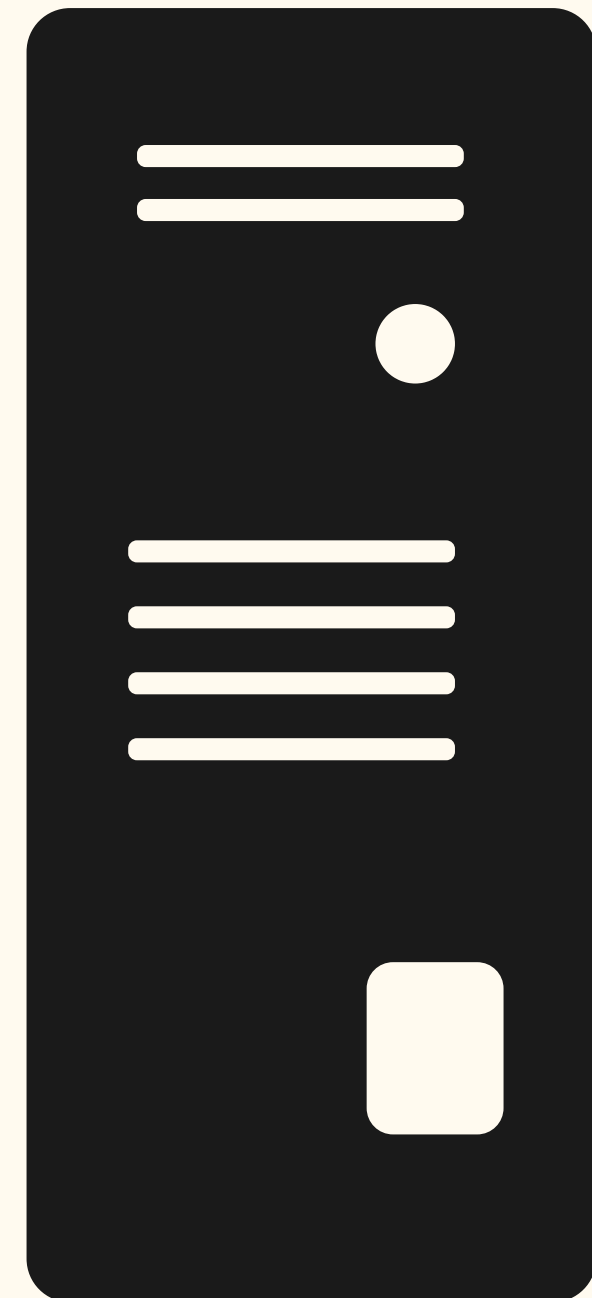
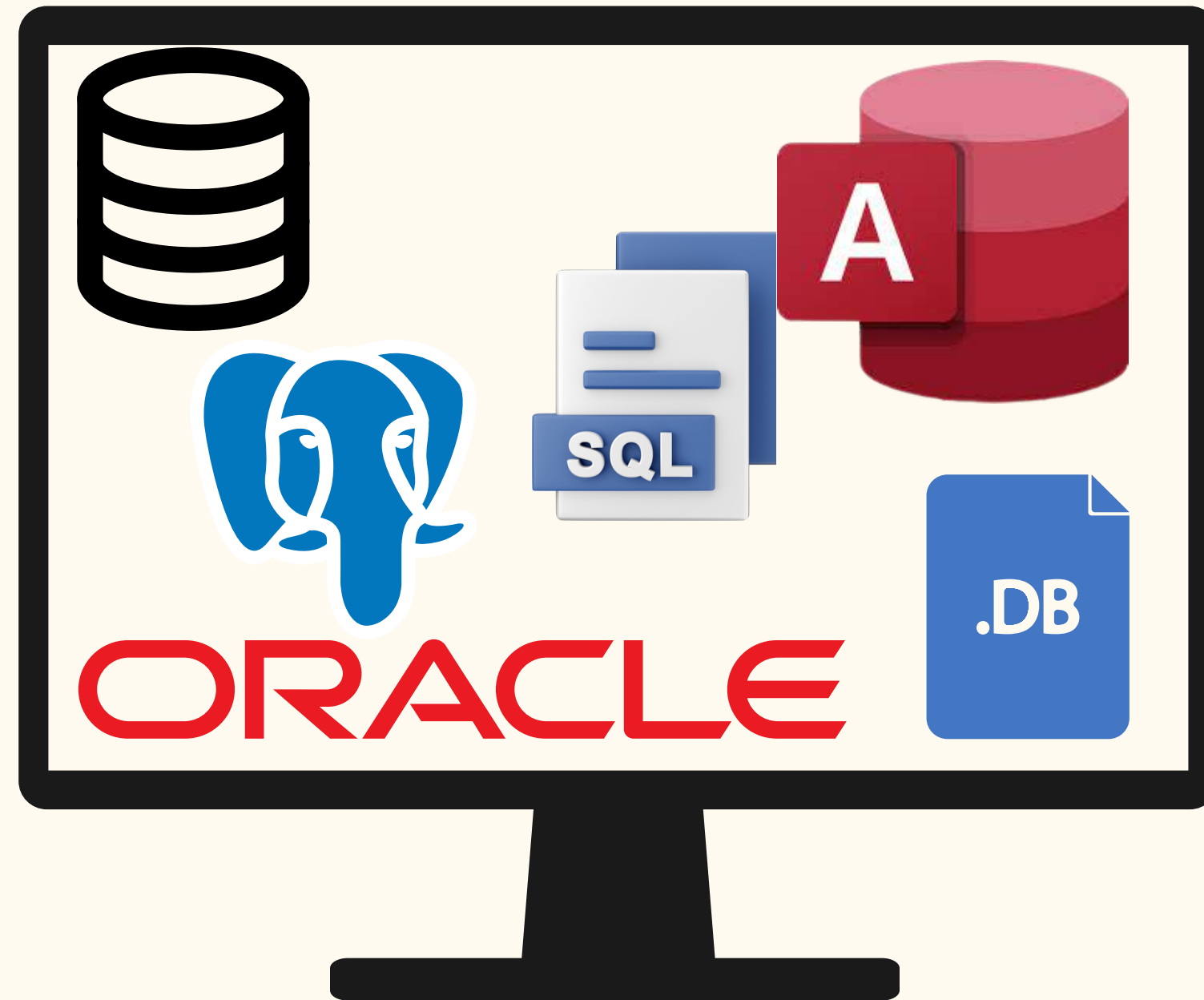
# Tìm kiếm trong cơ sở dữ liệu

Age ?

Name ?

Sex ?

Job ?



A large, abstract pink shape on the left side of the slide, resembling a stylized drop or a partial circle.

## **4. ƯU VÀ NHƯỢC ĐIỂM**

# Ưu điểm

- Tốc độ tìm kiếm nhanh
- Dễ cài đặt
- Độ phức tạp nằm trong khoảng từ  $O(\log N)$  đến  $O(N)$  trong đó  $N$  là số nút trong cây. (độ phức tạp thấp)



# Nhược điểm

- Hiệu suất kém khi số chiều dữ liệu tăng lên





## **5. Demo**

# Nguồn tham khảo

- <https://cs.brynmawr.edu/Courses/cs246/spring2013/slides/14KDTrees.pdf>
- <https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/kdtrees.pdf>
- <https://www.geeksforgeeks.org/search-and-insertion-in-k-dimensional-tree/>
- <https://www.geeksforgeeks.org/k-dimensional-tree-set-2-find-minimum/>