ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN – ĐIỆN TỬ

BÁO CÁO BÀI TẬP LỚN MÔN

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

Project 4: Priority Search Tree

Giảng viên hướng dẫn:	**Cô Tạ Kim Huệ**	
Sinh viên thực hiện:	**Nguyễn Thành Luân**	**20200375**
	Đỗ Tuấn Minh	**20200389**
	Lê Việt Anh	**20200020**
Lớp:	**CTTN-ĐTVT-K65**	
Mã lớp:	**139078**	

HÀ NỘI, NĂM 2023

LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, việc xử lý dữ liệu đang trở thành một phần quan trọng trong nhiều lĩnh vực, từ khoa học máy tính đến kinh doanh và y tế. Hiểu được cấu trúc dữ liệu và giải thuật sẽ giúp ta nắm bắt và xử lý thông tin một cách hiệu quả hơn.

Ppriority search tree là một trong những cấu trúc dữ liệu quan trọng và có vai trò quan trọng trong việc xử lý dữ liệu. Priority search tree được sử dụng để lưu trữ tập hợp các đối tượng có thứ tự và cho phép thực hiện các thao tác tìm kiếm, thêm, xóa đối tượng với độ phức tạp thời gian O(log n) trong trường hợp trung bình.

Với sự phát triển của các ứng dụng yêu cầu xử lý dữ liệu theo thứ tự ưu tiên, priority search tree được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm công nghệ thông tin, khoa học dữ liệu, và truy vấn trong cơ sở dữ liệu. Ví dụ như trong các ứng dụng tìm kiếm, nó có thể được sử dụng để lưu trữ và quản lý danh sách các kết quả tìm kiếm theo thứ tự ưu tiên.

Ngoài ra, priority search tree cũng được sử dụng trong các ứng dụng thời gian thực, ví dụ như trong các hệ thống điều khiển và tự động hóa. Trong các ứng dụng này, priority search tree có thể được sử dụng để quản lý các sự kiện được sắp xếp theo thời gian hoặc ưu tiên, giúp cho hệ thống có thể phản hồi nhanh chóng đến các sự kiện quan trọng.

Vì vậy, priority search tree đóng vai trò quan trọng trong việc xử lý dữ liệu và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau.

MỤC LỤC

I. Giới thiệu về Priority Search Tree:

Trong khoa học máy tính, cây tìm kiếm ưu tiên (priority search tree) là một cấu trúc dữ liệu dạng cây để lưu trữ các điểm trong không gian hai chiều (Oxy). Ban đầu cây tìm kiếm ưu tiên được giới thiệu bởi Edward McCreight năm 1985.

Hình 1.1 Edward McCreight

Ban đầu cây tìm kiếm ưu tiên là sự mở rộng của hàng đợi ưu tiên (priority queue) với mục đích cải thiện thời gian tìm kiếm từ:

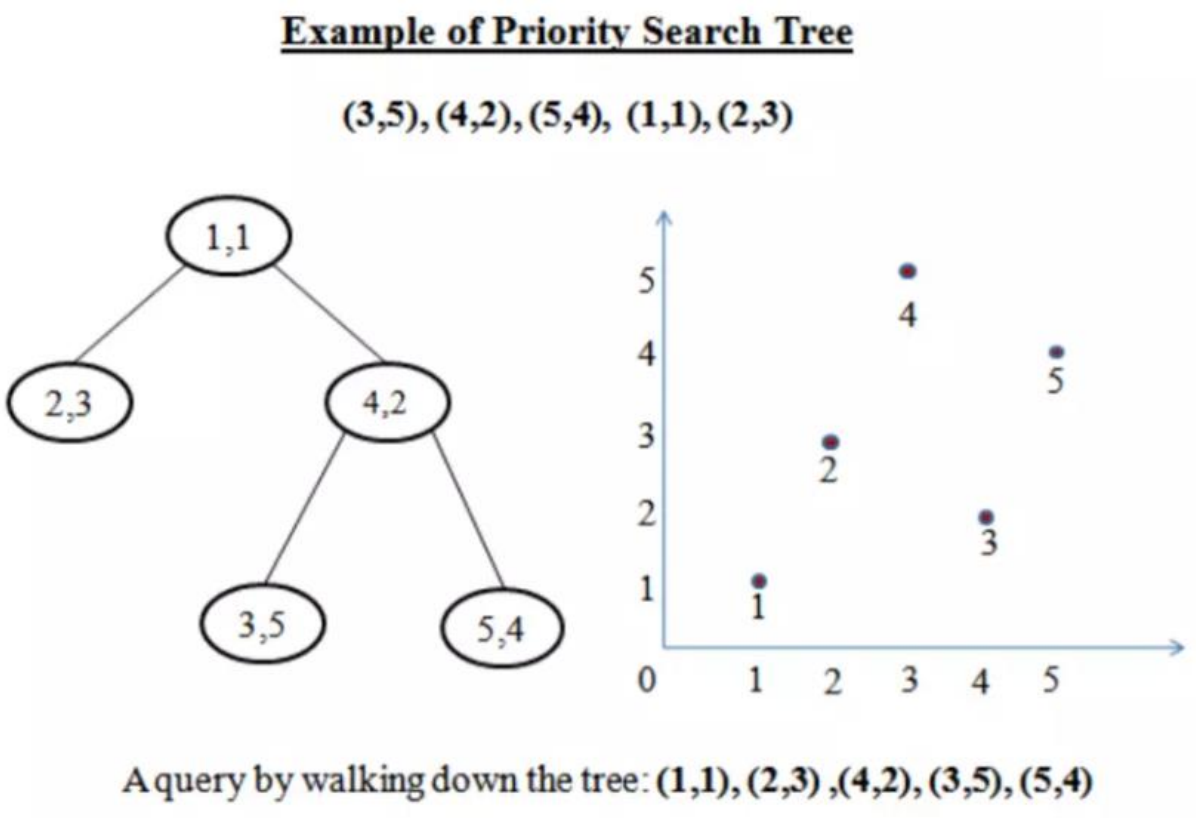
O(n) đến O(s+logn)

Trong đó n là số điểm trong cây và s là số trong tổng số điểm được trả về bởi tìm kiếm.

Sau đó, cây tìm kiếm ưu tiên được sử dụng để lưu trữ một tập hợp các điểm 2 chiều được sắp xếp theo mức độ ưu tiên(priority) và theo một giá trị khóa (key value). Điều này được thực hiện bằng cách tạo kết hợp giữa hàng đợi ưu tiên (priority queue) và cây tìm kiếm nhị phân (binary search tree).

Kết quả là một cây trong đó mỗi nút đại diện cho một điểm trong tập dữ liệu gốc. Điểm được chứa bởi nút là điểm có mức độ ưu tiên thấp nhất. Ngoài ra, mỗi nút còn chứa một giá trị khóa dùng để chia các điểm còn lại (thường là trung vị của các khóa, không kể điểm của

nút) thành cây con trái và phải. Các điểm được chia bằng cách so sánh các giá trị khóa của chúng với khóa nút, ủy nhiệm các giá trị có khóa thấp hơn cho cây con bên trái và các giá trị lớn hơn cho cây con bên phải.

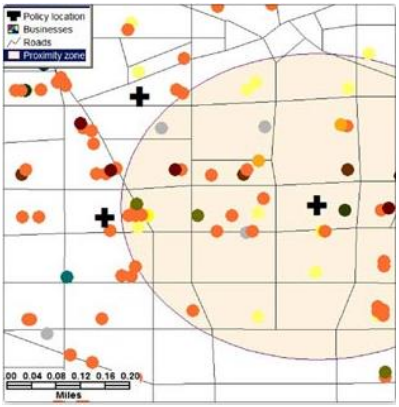


Hình 1.2 Ví dụ về một Priority Search Tree

II. Cấu trúc của Priority Search Tree:

1. Khái niệm:

Priority Search Tree (PST) là một cấu trúc dữ liệu được sử dụng trong lĩnh vực xử lý dữ liệu không gian đa chiều (spatial data processing). PST là một cây nhị phân cân bằng, mỗi nút trong cây chứa một trục và một danh sách các điểm dữ liệu. Các điểm dữ liệu được sắp xếp theo thứ tự tăng dần hoặc giảm dần trên trục tương ứng.



Hình 2.1 Ứng dụng của PST trong không gian đa chiều

Điểm mạnh của PST là giảm thiểu độ phức tạp của các phép toán tìm kiếm trong không gian đa chiều. PST cho phép tìm kiếm nhanh chóng các phần tử gần nhất hoặc các phần tử trong khoảng cách cố định trong không gian đa chiều. PST cũng cho phép tìm kiếm các phần tử lớn nhất hoặc nhỏ nhất trên một trục cụ thể trong không gian đa chiều.

Tuy nhiên, PST cũng có một số hạn chế. Khi số lượng điểm dữ liệu trong không gian đa chiều quá lớn, việc xây dựng PST có thể trở nên rất tốn kém và không hiệu quả. Ngoài ra, việc phân chia không gian dữ liệu vào các phần bằng nhau trên mỗi trục cũng không phải là phương pháp tối ưu cho mọi tình huống. Tuy nhiên, với việc thiết kế và tối ưu hóa phù hợp, PST có thể trở thành một công cụ hữu ích trong xử lý dữ liệu không gian đa chiều.

2. Cấu trúc Priority Search Tree (PST):

PST là một cây nhị phân cân bằng, trong đó mỗi nút là một trục và chứa một danh sách các điểm dữ liệu trong không gian đa chiều. Các điểm này được sắp xếp theo thứ tự tăng dần hoặc giảm dần trên trục tương ứng. Mỗi nút có thể có hai nút con, tương ứng với các khoảng giá trị trên trục tương ứng. Vì vậy, PST sẽ chia không gian đa chiều thành các phần con, mỗi phần con được liên kết với một nút của cây.

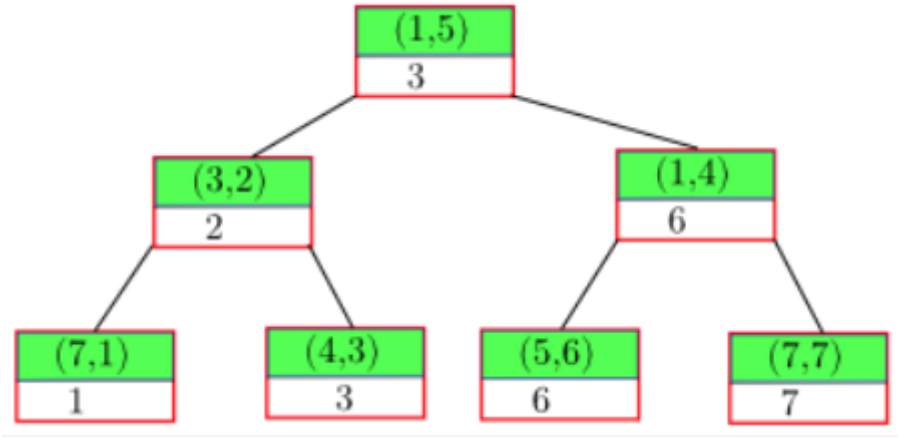
Với một tập hợp các điểm dữ liệu 2 được sắp xếp theo mức độ ưu tiên(priority) và theo một giá trị khóa (key value), sẽ sử dụng PST để lưu trữ:

- Mỗi nút đại diện cho một điểm trong tập dữ liệu gốc.
- Điểm được chứa bởi nút là điểm có mức độ ưu tiên thấp nhất.

Ngoài ra, mỗi nút còn chứa một giá trị khóa dùng để chia các điểm còn lại (thường là trung vị của các khóa, không kể điểm của nút) thành cây con trái và phải. Các điểm được

chia bằng cách so sánh các giá trị khóa của chúng với khóa nút, ủy nhiệm các giá trị có khóa thấp hơn cho cây con bên trái và các giá trị lớn hơn cho cây con bên phải.

Ví dụ cây PST với tập P như trên được biểu diễn ở hình sau. Các cặp giá trị ở phần background màu xanh của mỗi nút là một điểm biểu diễn bởi nút đó. Giá trị ở phần dưới là điểm chia y_v.



Hình 2.2 Ví dụ về cấu trúc cây

Ta có thể quan sát thấy việc tìm kiếm theo tọa độ x của các điểm được "tích hợp" vào cây thông qua tính chất Heap và tìm kiếm theo tọa độ y của các điểm được "tích hợp" thông qua tính chia đôi. Tính cân bằng đảm bảo chiều cao của cây là [log n] . Với các tính chất đó, ta có thể nhận thấy rằng PST là một cấu trúc phù hợp để tìm kiếm hai chiều với độ phức tạp của mỗi thao tác tìm kiếm là O(log n). Dễ thấy cấu trúc cây PST có bộ nhớ là O(n).

III. Xây dựng cây tìm kiếm ưu tiên (PST)

Cây tìm kiếm ưu tiên là một cấu trúc dữ liệu lưu trữ các điểm trên không gian 2 chiều. Đầu tiên, chúng ta xem xét n các điểm trên cùng một mặt phẳng, mỗi điểm đều có tọa độ x và tọa độ y biểu diễn trên hình 2. Bài toán xây dựng cây tìm kiếm ưu tiên được mô tả như sau:

Đầu vào:

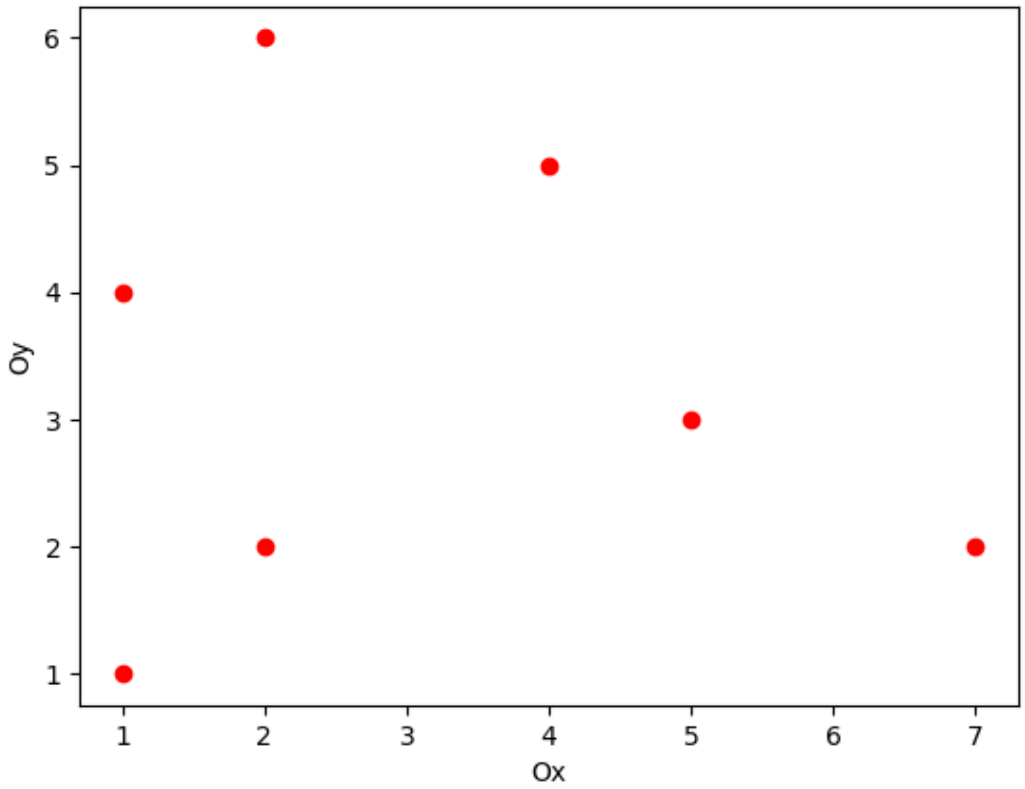
Ta có n điểm thuộc mặt phẳng $S = \{p_i \mid i = 1, 2, 3, 4, 5, \ldots n\}$ và các điểm $p_i\,(p_i.x, p_i.y)$

Đầu ra:

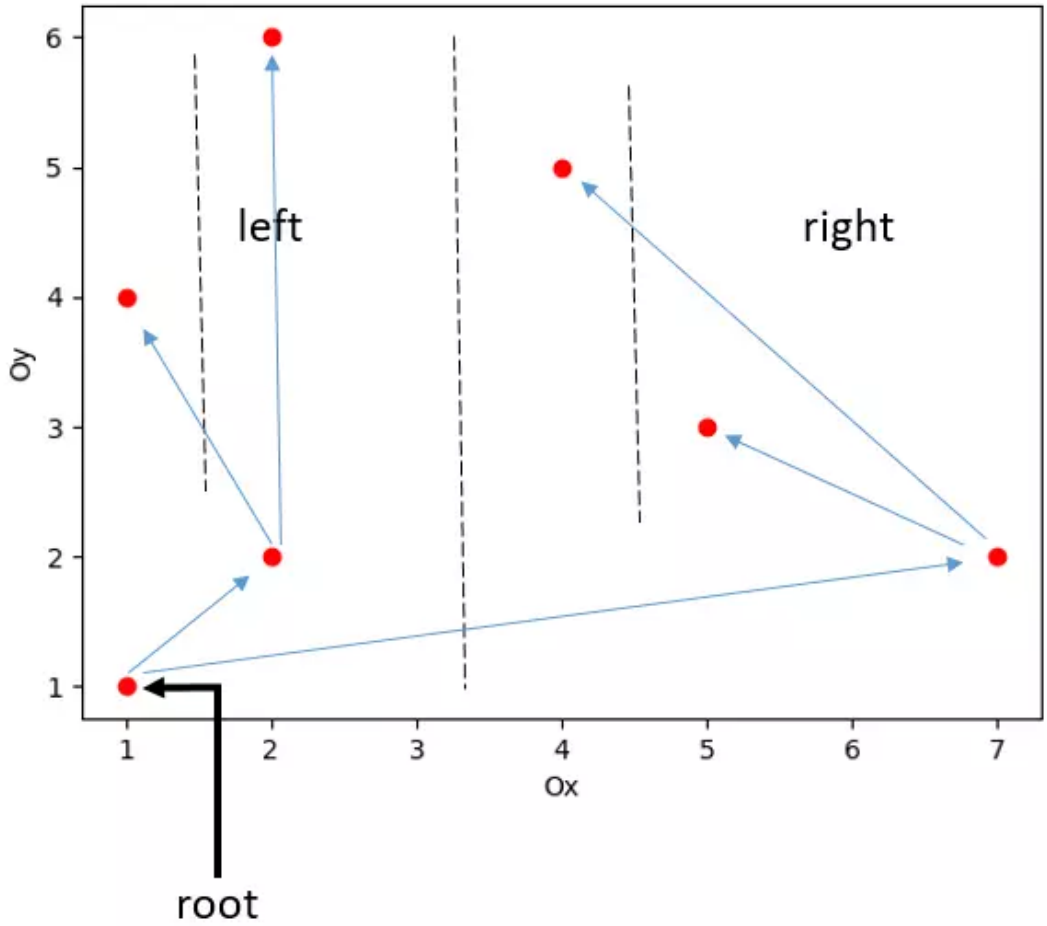
Cây tìm kiếm ưu tiên lưu trữ các điểm trong không gian 2 chiều.

Mã giả:

1. Nếu S = NULL thì trả về NULL.
2. Tìm điểm p_i có tọa độ nhỏ nhất làm root (gốc), $p_i = min\{ p_i.y \in SI\}$.
3. Tìm đường trung tuyến (median) chia đôi các điểm về 2 phía trái và phải. Ở phía bên phải tìm điểm $p_i = min\{ p_i .y \in S_l \backslash \{Parent\}\}$. *gán làm con của nút cha ở bước 2, thực hiện tương tự với phía bên phải xem hình 3.2* .
4. Đệ quy lại PST(s) với phía bên trái.
5. Đệ quy lại PST(s) với phía bên phải.
6. Kết thúc



Hình 3.1 Biểu diễn các điểm trên tọa độ Oxy



Hình 3.2 Mô tả cách xây dựng cây PST

IV. Các thao tác trên PST:

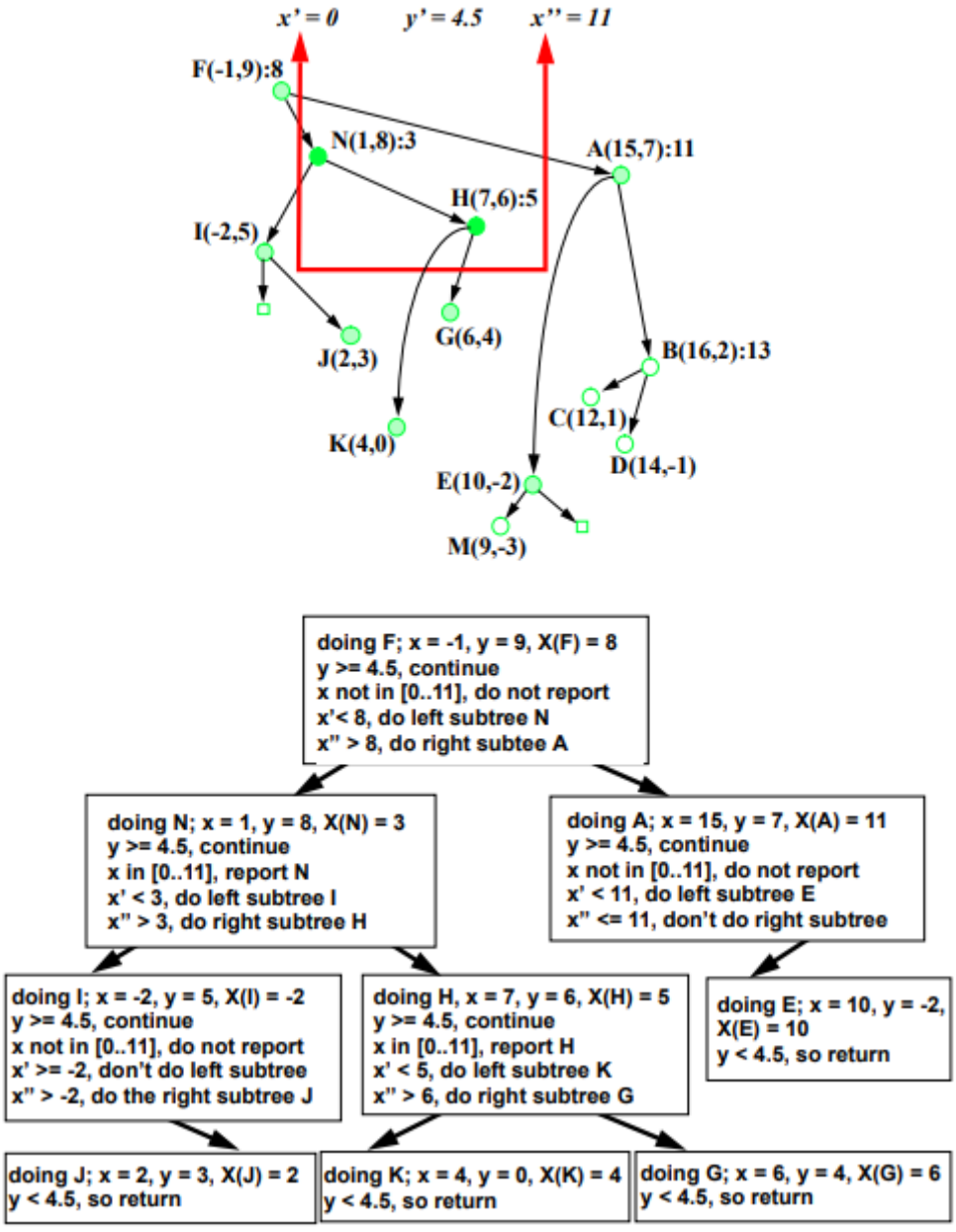
1. Truy vấn:

Một truy vấn trên cây tìm kiếm ưu tiên như sau: với x', x" và y', tìm tất cả các điểm trong cây có tọa độ X nằm giữa x' và x", và có tọa độ Y lớn hơn y' là gì? Sau đây là thuật toán để thực hiện truy vấn này:

- Nếu cây là NULL, chúng ta trả về mà không tìm thấy bất kỳ điểm nào.
- Hãy cho R là nút gốc của cây, x là tọa độ X của nó, y là tọa độ Y của nó và X(R) là giá trị của trục phân chia các khoảng X của các cây con của R.
- So sánh y với y' . Nếu y < y' , chúng ta trả về mà không tìm thấy bất kỳ điểm nào (tất cả các nút khác trong cây sẽ có tọa độ Y nhỏ hơn).
- Nếu **x' ≤ x ≤ x''** , báo cáo điểm gốc.
- Nếu **x' < X(R),** khoảng X của cây con bên trái phải chồng lên khoảng X của truy vấn. Đệ quy tìm kiếm cây con bên trái của R.

- Nếu X(R) < x 00 , khoảng X của cây con bên phải phải chồng lên khoảng X của truy vấn. Đệ quy tìm kiếm cây con bên phải của R.

Một ví dụ về truy vấn trên cây tìm kiếm ưu tiên, với x 0 = 0, x 00 = 11, y 0 = 4.5, được hiển thị trong Hình 3.3. Các nút được truy cập nhưng không được báo cáo được tô màu; các nút được truy cập và được báo cáo được tô đậm.



Hình 3.3 Tìm kiếm trên PST

2. Thêm một node mới:

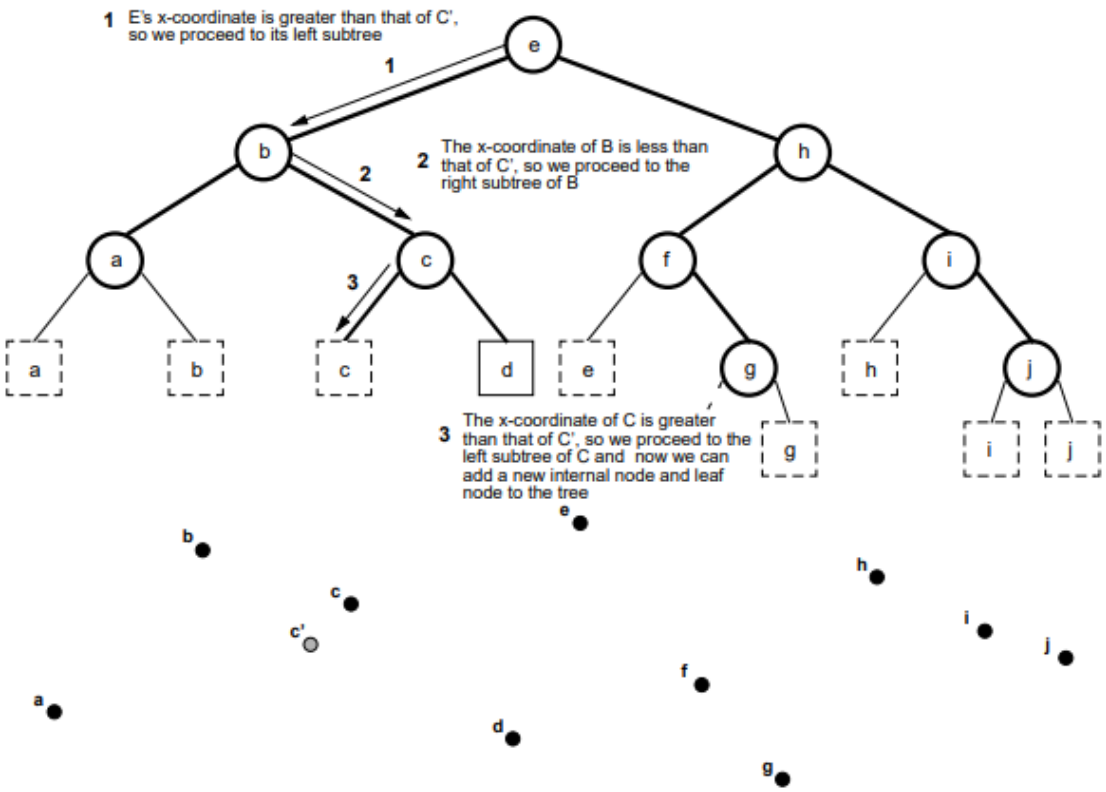
Để thêm một điểm vào cây tìm kiếm ưu tiên, ta thực hiện các bước sau:

- Thêm một lá mới (làm giữ chỗ) cho điểm mới

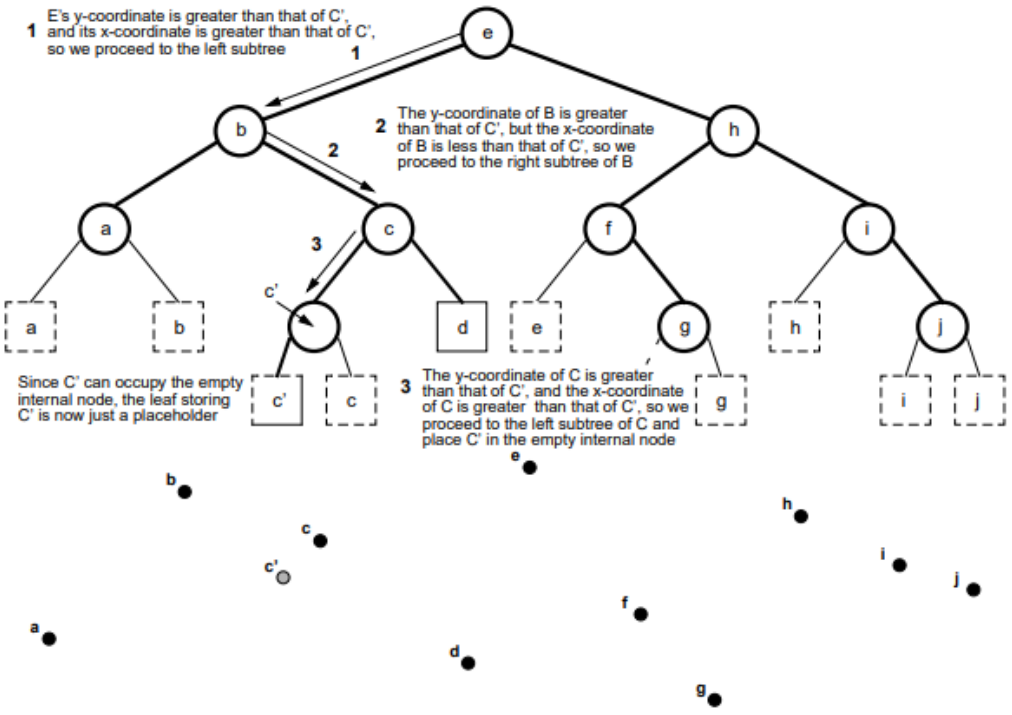
- Thực hiện phép đẩy xuống với điểm mới bắt đầu từ gốc
- Thực hiện cấu trúc lại để duy trì cân bằng của cây

Trong Hình 3.4, chúng ta thấy rằng một điểm mới c’ đã được thêm vào tập dữ liệu ban đầu của chúng ta. Bước đầu tiên của chúng ta sẽ là thêm một giữ chỗ mới cho điểm này, như được hiển thị trong Hình 3.5 .

Hình 3.4 Thêm nốt mới c’



Hình 3.5 Tìm kiếm trên cây để thêm vào lá có node c’



Hình 3.6 Kết quả sau khi thêm c'

V. Ứng dụng của PST:

Priority search trees có rất nhiều ứng dụng trong lĩnh vực xử lý đồ thị, tìm kiếm, truy vấn phạm vi và các vấn đề tương tự. Dưới đây là một số ứng dụng phổ biến của priority search trees:

1. Tìm kiếm phạm vi: Priority search trees được sử dụng để tìm các đối tượng trong phạm vi cho trước. Ví dụ: tìm các nhà hang gần nhất trên bản đồ.
2. Tìm kiếm giao điểm phạm vi: Priority search trees cũng được sử dụng để tìm giao điểm của các đoạn trong phạm vi cho trước. Ví dụ: tìm các điểm chung giữa các đoạn trong một tập hợp đoạn.
3. Các vấn đề xử lý đồ thị: Priority search trees được sử dụng để xử lý các vấn đề như tìm kiếm đường đi ngắn nhất và tìm kiếm đường đi tối ưu giữa các đỉnh của đồ thị.
4. Truy vấn hình học: Priority search trees cũng được sử dụng để giải quyết các vấn đề truy vấn hình học như tìm kiếm các điểm trong một hình chữ nhật, tìm kiếm các điểm gần nhất và tìm kiếm tất cả các điểm trong một vùng cho trước.

5. Truy vấn cơ sở dữ liệu: Priority search trees được sử dụng để xây dựng các cây dữ liệu cho việc truy vấn cơ sở dữ liệu và các tác vụ tìm kiếm phức tạp khác.
6. Nhiều ứng dụng khác: Priority search trees cũng được sử dụng trong các lĩnh vực như tin học đám mây, khoa học dữ liệu, học máy và nhiều lĩnh vực khác.

VI. KẾT LUẬN

Priority Search Tree, một cây nhị phân được sử dụng để giải quyết các truy vấn giao thoa khoảng, đặc biệt là tìm kiếm giao thoa của các khoảng trên đường thẳng.

Nhóm em đã tìm hiểu và nghiên cứu về việc xây dựng cây Priority Search Tree, đưa ra thuật toán xử lý cho việc chèn và xóa một khoảng vào cây. Một số ứng dụng của cây Priority Search Tree, bao gồm tìm kiếm giao thoa khoảng trên đường thẳng, tìm kiếm khoảng gần nhất, và xử lý các truy vấn tìm kiếm liên quan đến đồ thị.

Tuy nhiên, nhóm em cũng nhận thấy rằng Priority Search Tree không phải là giải pháp tốt nhất cho tất cả các bài toán. Một số bài toán có thể được giải quyết hiệu quả hơn bằng các cấu trúc dữ liệu khác.

Tóm lại, Priority Search Tree là một cấu trúc dữ liệu quan trọng trong việc giải quyết các bài toán truy vấn giao thoa khoảng trên đường thẳng và đồ thị. Việc xây dựng và sử dụng cây Priority Search Tree tĩnh và động là rất hữu ích trong nhiều ứng dụng khác nhau.

Cuối cùng, nhóm em xin cảm ơn cô Tạ Kim Huệ đã hướng dẫn và hỗ trợ chúng em trong quá trình tìm hiểu về priority search tree cũng như các kiến thức liên quan đến cấu trúc dữ liệu và giải thuật.

TÀI LIỆU THAM KHẢO

1. scribe.dvi (brown.edu)

2. Cây tìm kiếm ưu tiên – Wikipedia tiếng Việt

3. Tìm Hiểu Về Cấu Trúc Dữ Liệu Cây Tìm Kiếm Ưu Tiên (Priority Search Tree) (viblo.asia)

4. Edward M. McCreight, "Priority Search Trees," SIAM J. Comput., Vol. 14, No. 2, pp. 257–276, May 1985

5. Alok N. Choudhary, "Dynamic Priority Search Trees," November 1987..