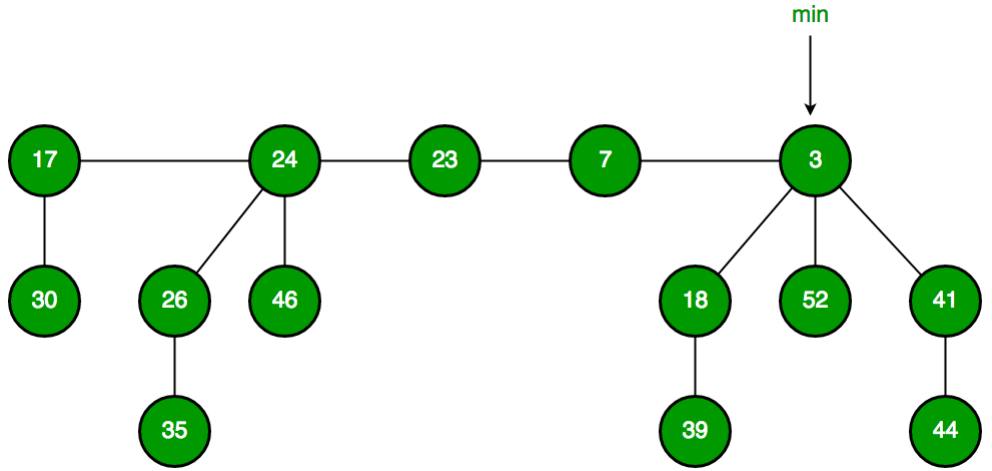
Bài 7.7: Fibonacci heap

- ✓ Khái niệm
- ✓ Cấu trúc
- ✓ Các thao tác trên Fibonacci heap

Khái niệm

- Fibonacci heap là cấu trúc dữ liệu sử dụng chủ yếu trong thực hiện các hành động của hàng đợi ưu tiên.
- Nó bao gồm tập hợp các heap tuân thủ quy tắc min heap.
- Có thời gian chạy tốt hơn các cấu trúc dữ liệu dùng để tạo hàng đợi ưu tiên khác như đống nhị phân, đống nhị thức.



Cấu trúc của đống Fibonacci

- Fibonacci heap có một con trỏ min trỏ đến node có giá trị nhỏ nhất trong đống.
- Tất cả các node gốc của các cây trong đống Fibonacci được liên kết với nhau qua danh sách liên kết đôi vòng. Do đó ta có thể sử dụng con trỏ min để duyệt đến tất cả các cây trong đống.
- Đống Fibonacci mềm dẻo hơn đống Binomial vì nó cho phép các cây có số lượng node tùy ý.
- Điều này làm cho một số hành động của heap được phép thực hiện theo kiểu lười biếng, trì hoãn hành động cho đến các hành động sau đó.
- Ví dụ: việc trộn hai đống được thực hiện đơn giản bằng cách liên kết hai cây lại với nhau. Hay hành động thêm một node vào heap được thực hiện bằng cách tạo mới cây với 1 node và trộn cây đó vào đống.

Cấu trúc của đống Fibonacci

- Hành động xóa phần tử nhỏ nhất là một trong những hành động khó nhất của đống Fibonacci.
- Đống được gọi là đống Fibonacci vì số Fibonacci được sử dụng trong phân tích thời gian chạy nó.
- Mặt khác các cây trong đống có đặc điểm: mỗi node số node con trực tiếp tối đa là logn, kích thước của cây con có gốc là node có độ k là F(k+2) trong đó F(k) là số Fibonacci thứ k.

Các thao tác

- Các node trong cây cấu thành đống Fibonacci được liên kết với nhau bằng danh sách liên kết. Tại mỗi node ta sẽ quản lý số node con của nó.
- Merge:
  - Được thực hiện bằng cách nối hai node gốc của hai heap lại với nhau.
  - Độ phức tạp của thao tác merge là O(1).
- Insert:
  - Thêm mới phần tử vào đống có thể thực hiện bằng cách tạo một đống mới chứa duy nhất 1 phần tử và trộn nó vào đống trước đó.
  - Thao tác insert có độ phức tạp O(1).

Các thao tác

- Tìm node nhỏ nhất trong đống:
  - Sử dụng con trỏ min giữ liên kết đến node có giá trị nhỏ nhất trong heap.
  - Độ phức tạp O(1).
- Xóa node nhỏ nhất gồm 3 pha:
  - Đầu tiên xóa node nhỏ nhất khỏi gốc của cây nhị thức chứa nó. Sau đó tái cân bằng cây có node bị xóa.
  - Pha thứ 2 tiến hành giảm số lượng node gốc trong đống bằng cách liên kết các node gốc cùng độ lại. Khi liên kết hai gốc cùng độ, node nào có giá trị nhỏ hơn sẽ là cha, node còn lại là node con. Pha này được thực hiện tới khi mọi node gốc trong đống Fibonacci có độ khác nhau.
  - Pha thứ 3 tiến hành tìm node nhỏ nhất trong đống.
  - Độ phức tạp tổng thể O(logn).

Các thao tác

- Cập nhật giá trị một node trong đống:
  - Thực hiện rất loằng ngoằng phức tạp.
  - Độ phức tạp thuật toán: O(1).
- Xóa một node bất kì:
  - Để xóa node bất kì khỏi đống nhị thức, ta giảm giá trị của node đó xuống âm vô cực hoặc giá trị nhỏ nhất trong đống chứa nó và tiến hành xóa node nhỏ nhất trong đống(đã mô tả ở phần trước).
  - Độ phức tạp O(logn).

Nội dung tiếp theo

Binomial heap