Bài 7.8: Binomial heap

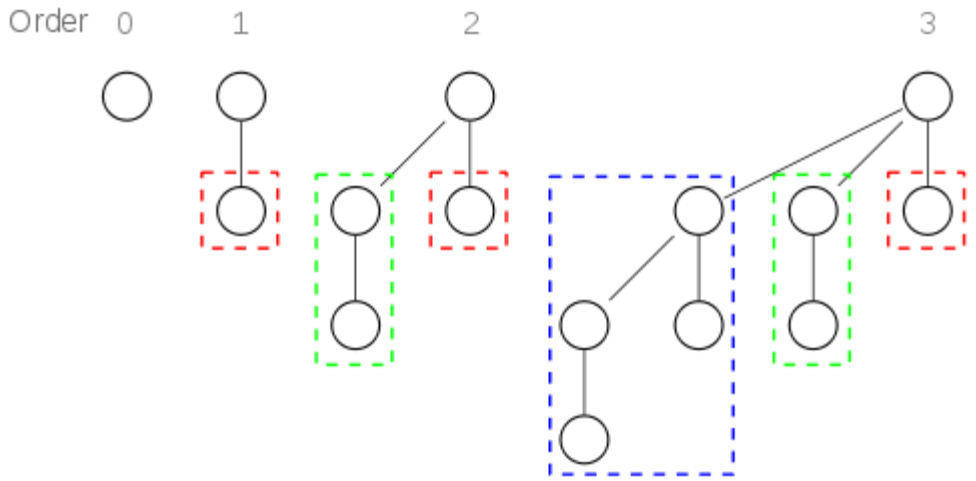
- ✓ Khái niệm
- ✓ Cấu trúc
- ✓ Các thao tác trên binomial heap

Khái niệm

- Một binomial heap tức đống nhị thức là một cấu trúc dữ liệu hoạt động như hàng đợi ưu tiên. Ngoài ra nó còn cho phép ghép nối với các đống nhị thức khác để tạo đống nhị thức lớn hơn.
- Đống nhị thức được tạo thành từ một tập các cây nhị thức(có dạng các cây nhị phân đơn) được định nghĩa một cách đệ quy như sau:
  - Một cây nhị thức bậc 0 có 1 node.
  - Một cây nhị thức bậc k có node gốc liên kết đến node con là các node gốc của các cây nhị thức bậc k-1, k-2, … 0.
- Cây nhị thức bậc k có 2^k node và có độ cao k.
- Cây nhị thức bậc k có thể được tạo thành từ hai cây nhị thức bậc k-1, trong đó một cây sẽ là cây con trái cùng của cây còn lại. Đây chính là tâm điểm của sự hợp nhất đống nhị phân.

Khái niệm

- Ví dụ minh họa:



Cấu trúc của đống nhị thức

Đống nhị thức được triển khai bằng cách liên kết các cây nhị thức lại thành một tập hợp thỏa mãn các tính chất của đống nhị thức:

- Mỗi cây nhị thức trong heap tuân thủ tính chất của min heap, tức node cha có giá trị nhỏ hơn hoặc bằng các node con của nó.
- Chỉ có thể có tối đa 1 cây nhị thức mỗi bậc trong đống nhị thức, bao gồm cả bậc 0.

Tính chất đầu tiên đảm bảo rằng gốc của mỗi cây nhị thức chứa khóa nhỏ nhất trong cây.

Tính chất thứ 2 ngụ ý rằng mỗi đống nhị thức với n node chứa tối đa 1 + ln(n) cây nhị thức.

Các thao tác

- Ta thường lưu gốc của mỗi cây nhị thức trong đống nhị thức vào danh sách liên kết. Sắp xếp theo trật tự bậc của cây.
- Merge:
  - Thường được sử dụng để tạo ra một heap nhị thức mới.
  - Trong đó cây nhị thức có khóa lớn hơn trong 2 cây sẽ trở thành cây con của cây còn lại.
  - Độ phức tạp của thao tác merge là O(logn).
- 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(logn).

Các thao tác

- Tìm node nhỏ nhất trong đống nhị thức:
  - Tiến hành so sánh các node gốc của các cây trong đống để tìm ra kết quả.
  - Độ phức tạp của việc tìm node nhỏ nhất là O(logn).
  - Nếu sử dụng con trỏ min thì độ phức tạp chỉ còn O(1).
- Xóa node nhỏ nhất:
  - Đầu tiên phải tìm được node nhỏ nhất trong đống.
  - Tiếp theo xóa nó khỏi gốc của cây nhị thức chứa nó.
  - Sau đó tách các node còn lại của cây chứa node nhỏ nhất ra tạo thành một min heap mới đặt là tmp.
  - Xóa bỏ cây chứa node nhỏ nhất khỏi đống nhị thức.
  - Trộn cây tmp vào đống nhị thức.
  - Độ phức tạp O(logn).

Các thao tác

- Cập nhật giá trị một node trong đống:
  - Sau khi cập nhật ta sẽ phải tái cân bằng cây chứa node được cập nhật.
  - Thao tác này yêu cầu phải tìm được cây nhị thức chứa node cần cập nhật, do đó sẽ phức tạp(khó thực hiện) hơn các thao tác khác.
  - Độ phức tạp thuật toán: O(logn).
- 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

Một số hàm thao tác với heap trong C++