**Heap**

* Bắt đầu với cây nhị phân, được tạo lập bằng class và con trỏ.
* Binary heap:
  + Áp dụng trên cây nhị phân dạng hoàn chỉnh: Nút đỉnh có giá trị lớn hơn hoặc nhỏ hơn nút con (min/max heap)
  + Vì đã sắp xếp nên sẽ đc lưu dưới dạng mảng
* Thêm nút: Giả sử có một min/max heap tree, muốn thêm một phần tử bất kỳ:
  + Đầu tiên thêm phần tử vào cuối cây
  + Điều chỉnh lần lượt vị trí bằng cách so sánh với nút cha và điều chỉnh
  + Đọ phức tạp O(n)

H P E X A M P

P

H E

XA MP

P

X E

HA MP

X

P E

HA MP

X

P P

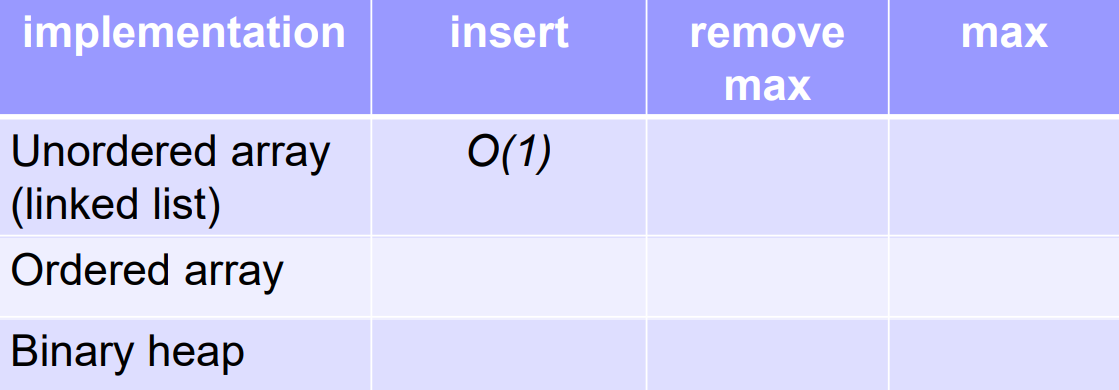
HA EM

* Xóa nút đỉnh:
  + Loại bỏ nút đỉnh
  + Thay thế bằng nút cuối cùng (để cây cân bằng)
  + Tạo cây heap bằng cách cũ
* Dìm xuống(sink down):
  + Xuất phát từ gốc
  + So sánh với các nút con, đổi chỗ với nút con lớn hơn( để đảm bảo thứ tự)

**Tìm trường hợp thay thế node xóa bằng lá không nằm cùng nhánh???????????????????**

**PRIORITY QUEUE**

* Cài đặt
  + Bằng mảng: Sắp xếp mảng
  + Bằng linked list: dễ insert
  + Heap??



O(logn)

O(logn)

O(n))

1

1

1

O(n)

O(n)