

Bài 0:

Cài đặt lại bài kiểm tra về Cấu trúc dữ liệu Set.

Bài 1:

Dựa vào bài số 7 của bài thực hành trước, viết chương trình sắp xếp bằng thuật toán sắp xếp hòa nhập (Merge Sort).

Vẽ cây mô tả lời gọi đệ quy với mảng đầu vào có 6 phần tử.

Bài 2:

Dựa vào bài số 8 của bài thực hành trước, viết chương trình sắp xếp bằng thuật toán sắp xếp nhanh (Quick Sort).

Vẽ cây mô tả lời gọi đệ quy với mảng đầu vào có 6 phần tử.

Bài 3:

Sử dụng mảng để biểu diễn Heap có thứ tự theo mô tả sau:

- Nút cha không nhỏ hơn 2 nút con
- Chỉ số nút trên mảng bắt đầu từ 1
- Nút thứ i sẽ có 2 nút con là $2i$ và $2i+1$

Viết các phương thức để xây dựng Heap có thứ tự:

- Phương thức `upheap(int k)`, phương thức này sử dụng khi nút ở vị trí thứ k lớn hơn nút cha ở vị trí $k/2$
 - o Phương thức `upheap` thực hiện việc đổi chỗ phần tử hiện tại (vị trí k) với phần tử ở nút cha của nó (vị trí $k/2$), việc đổi chỗ thực hiện cho đến khi gặp nút gốc ($k = 1$) hoặc nút hiện tại nhỏ hơn nút cha
 - Phương thức `upheap` được gọi tới khi 1 phần tử mới được thêm vào mảng
- Phương thức `insert` thực hiện thêm 1 phần tử vào mảng
 - o Đưa phần tử mới vào mảng và gọi phương thức `upheap` để vun lại Heap, đảm bảo tính thống nhất của Heap
- Phương thức `downheap(int k)`, phương thức này sử dụng khi nút ở vị trí thứ k nhỏ hơn nút con của nó ở vị trí $2k$ và $2k+1$
 - o Phương thức này thực hiện việc đổi chỗ nút hiện tại ở vị trí k cho nút con lớn hơn trong 2 nút con

ở vị trí $2k$ và $2k+1$, việc đổi chỗ thực hiện cho đến khi nút k là nút lá ($2k > n$) hoặc nút k lớn hơn cả 2 nút con

- Phương thức sort thực hiện sắp xếp mảng gồm n phần tử
 - o Phương thức sort thực hiện việc đổi chỗ phần tử ở nút gốc (vị trí 1) với phần tử cuối của mảng, sau đó thực hiện downheap ở vị trí số 1 với $n-1$ phần tử để tạo lại heap. Việc đổi chỗ và downheap thực hiện cho đến khi số phần tử của heap = 1.

Bài 4:

Thực hiện đánh giá thời gian chạy của các thuật toán trên với các dãy đầu vào khác nhau.