CÂU HỎI CHƯƠNG 3

3)

 Thuật toán merge sort

Giống như Quick sort, Merge sort là một thuật toán chia để trị.

Thuật toán này: - chia mảng cần sắp xếp thành 2 nửa.

* Tiếp tục lặp lại việc này ở các nửa mảng đã chia.
* -Sau cùng gộp các nửa đó thành mảng đã sắp xếp.

+ Hàm merge() được sử dụng để gộp hai nửa mảng. Hàm merge(arr, l, m, r) là tiến trình quan trọng nhất sẽ gộp hai nửa mảng thành 1 mảng sắp xếp, các nửa mảng là arr[l…m] và arr[m+1…r] sau khi gộp sẽ thành một mảng duy nhất đã sắp xếp.

mergeSort(arr[], l, r)

If r > l

1. Tìm chỉ số nằm giữa mảng để chia mảng thành 2 nửa:

middle m = (l+r)/2

2. Gọi đệ quy hàm mergeSort cho nửa đầu tiên:

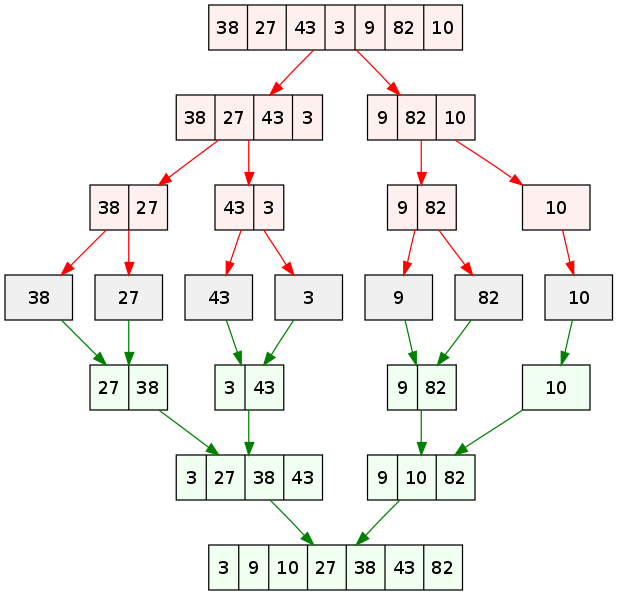
mergeSort(arr, l, m)

3. Gọi đệ quy hàm mergeSort cho nửa thứ hai:

mergeSort(arr, m+1, r)

4. Gộp 2 nửa mảng đã sắp xếp ở (2) và (3):

merge(arr, l, m, r)



**Độ  phức tạp thuật toán**

* Trường hợp tốt: O(nlog(n))
* Trung bình: O(nlog(n))
* Trường hợp xấu: O(nlog(n))

Không gian bộ nhớ sử dụng: O(n)

1. Phương pháp tìm kiếm Quicksort là phương pháp tối ưu nhất .Vì là thuật toán chạy nhanh nhất trong trường hợp trung bình, không cần bộ nhớ phụ , xài tốt cho DSLK.

Phương pháp tìm kiếm Buble Sort là kém tối ưu nhất là không nhận diễn được tình trạng dãy đã có thứ tự từng phần. Các phần tử nhỏ được đưa về vị trí đúng rất nhanh , trong khi các phần tử lớn lại được đưa về vị trí đúng rất chậm

1. 2 phương pháp tìm kiếm tuần tự và tìm kiếm nhị phân là như nhau nếu đã sắp xếp sẵn thứ tự.