HỆ ĐIỀU HÀNH
TUẦN 9
QUẢN LÝ TIỂU TRÌNH (LUỒNG - THREAD)

Làm việc với tập tin

```
FILE *ptr;
ptr = fopen("fileopen","mode");
```

For example,

```
fopen("E:\\cprogram\\newprogram.txt","w");

fopen("E:\\cprogram\\oldprogram.txt","r");
```

```
fclose(ptr);
```

Bài tập ứng dụng 1:

Cho một tập tin input.txt có cấu trúc sau:
- Dòng đầu tiên chứa số phần tử mảng
- Dòng còn lại chứa các phần tử là số nguyên

Viết chương trình gồm các thread thực hiện các công việc sau:
- Thread thứ nhất đọc file đầu vào là đối số thứ nhất từ biến môi trường
- Thread thứ hai tính tổng các số nguyên tố trong mảng
- Thread thứ ba tính sắp xếp mảng tăng dần
- Thread thứ tư thực hiện việc ghi file result.txt

Nội dung file đầu vào và đầu ra như sau:
Ví dụ:
File đầu vào input.txt có dạng sau:

```
10
4 5 7 8 11 9 20 13 2 3
```

Kết quả trong file result.txt có dạng sau:

```
So phan tu mang: 10
4 5 7 8 11 9 20 13 2 3
Mang cac so nguyen to:
5 7 11 13 2 3
Tong cac so nguyen to: 41
Mang cac so nguyen to da duoc sap xep 2 3 5 7 11 13
```

Ý tưởng thuật toán Merger sort

- 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.

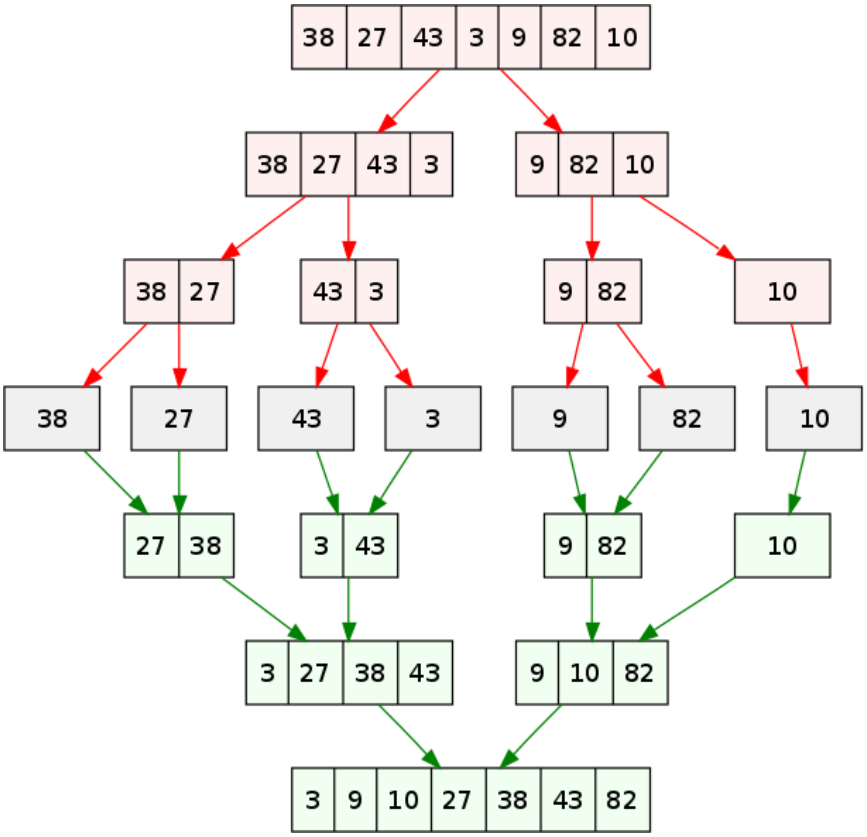
Hàm merge(arr, l, m, r) là tiến trình quan trọng nhất sẽ gộp hai nửa mảng thành 1 mảng sắp xếp, các nửa mảng là arr[l…m] và arr[m+1…r] sau khi gộp sẽ thành một mảng duy nhất đã sắp xếp.

Hãy xem ý tưởng triển khai code dưới đây để hiểu hơn

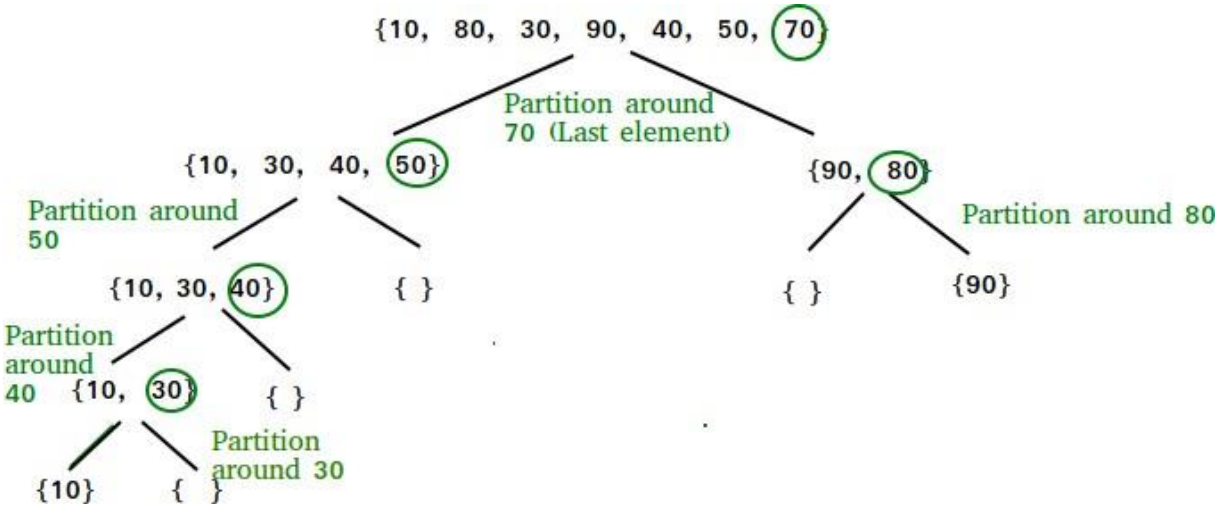
```
0
1  mergeSort(arr[], l, r)
2  If r > l
3       1. Tìm chỉ số nằm giữa mảng để chia mảng thành 2 nửa:
4            middle m = (l+r)/2
5       2. Gọi đệ quy hàm mergeSort cho nửa đầu tiên:
6            mergeSort(arr, l, m)
7       3. Gọi đệ quy hàm mergeSort cho nửa thứ hai:
8            mergeSort(arr, m+1, r)
9       4. Gộp 2 nửa mảng đã sắp xếp ở (2) và (3):
10           merge(arr, l, m, r)
11
```

Hình ảnh dưới đây từ wikipedia sẽ hiển thị cho bạn toàn bộ sơ đồ tiến trình của thuật toán merge sort cho mảng {38, 27, 43, 3, 9, 82, 10}.

Nếu nhìn kỹ hơn vào sơ đồ này, chúng ta có thể thấy mảng ban đầu được lặp lại hành động chia cho tới khi kích thước các mảng sau chia là 1. Khi kích thước các mảng con là 1, tiến trình gộp sẽ bắt đầu thực hiện gộp lại các mảng này cho tới khi hoàn thành và chỉ còn một mảng đã sắp xếp.



Ý tưởng của thuật toán sắp xếp Quick sort



Mô phỏng thuật toán sắp xếp quick sort

Giống như Merge sort, thuật toán sắp xếp quick sort là một thuật toán chia để trị(Divide and Conquer algorithm). Nó chọn một phần tử trong mảng làm điểm đánh dấu(pivot). Thuật toán sẽ thực hiện chia mảng thành các mảng con dựa vào pivot đã chọn. Việc lựa chọn pivot ảnh hưởng rất nhiều tới tốc độ sắp xếp. Nhưng máy tính lại không thể biết khi nào thì nên chọn theo cách nào. Dưới đây là một số cách để chọn pivot thường được sử dụng:

Luôn chọn phần tử đầu tiên của mảng.

Luôn chọn phần tử cuối cùng của mảng. (Được sử dụng trong bài viết này)

Chọn một phần tử random.

Chọn một phần tử có giá trị nằm giữa mảng(median element).

Bài tập ứng dụng 2:

Tạo 1 mảng các phần tử chưa được sắp xếp. Hãy xây dựng 1 hàm sắp xếp dựa trên các thuật toán đã học. Hãy tách mảng này thành 2 mảng con. Sau đó, với mỗi mảng con, gọi 1 tiểu trình thực hiện sắp xếp mảng đó. Sau khi thực hiện xong, thực hiện nối 2 mảng này.