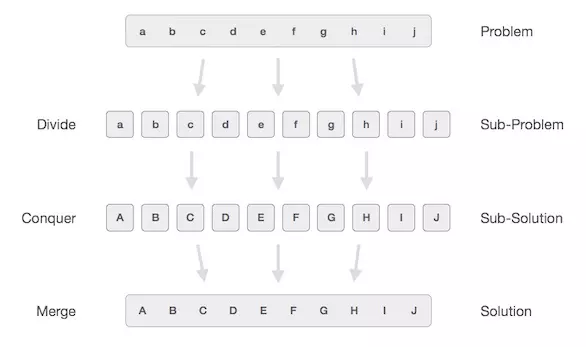
Chia để trị.Quick Sort.Binary Search.

Khái quát chung về chia để trị

Khái niệm: Là phương pháp chia nhỏ bài toán con,giải quyết chúng và tổng hợp thành bài toán ban đầu



Các bước giải

B1:Chia nhỏ thành các bài toán con đến mức không chia được nữa

B2:Giải quyết bài toán con

B3:Kết hợp lời giải lại để ra kết quả

Vd:Tính 2^n

-Thuật toán cơ bản: chạy vòng for(1->n) (ĐPT:O(n))

-Thuật toán chia để trị: (ĐPT: O(log(n)) )

+Ta chia 2^n  thành: 2^n =2^(n/2) \*2^(n/2)(nếu n chẵn)

                                             =2\*(2^n/2)\* (2^n/2) (nếu n lẻ )

       +Tiếp tục chia như thế đến khi n=0==> 2^n=1; n=1==> 2^n=2;

QuickSort

1.KN:Là 1 thuật toán sắp xếp sử dụng kĩ thuật chia để trị

2.Mô tả thuật toán :

+Nếu mảng còn 1 phần tử thì không phải sắp xếp

+Chia:

Chọn 1 phần tử trong dãy làm “PIVOT” hay 1 chốt

Chia đoạn đã cho thành 2 dãy con:-Bên trái(L) chốt là các phần tử nhỏ hơn chốt

    -Bên phải(R) chốt là các phần tử lớn hơn chốt

(Thường thì sẽ chọn chốt: ngoài cùng bên phải;trái;chính giữa;phần tử trung vị của phần tử đứng ngoài cùng bên trái,phải,giữa; và cũng có thể là phần tử ngẫu nhiên)

+) NOTE:Phần tử Pivot :

-Độ phức tạp thuật toán QuickSort :Từ O(nlog(n)) đến O(n^2) tuỳ vào việc chọn Pivot.

   +O(n^2) :Đây là trường hợp xấu nhất, khi dãy cho đã được sắp xếp và ta lại xét từ đầu dãy (Phân đoạn không cân bằng)

   +O(nlogn) :Khi phân đoạn,các đoạn nhỏ có kích thước luôn bằng nhau (Mỗi lần chia là chia đôi) (Phân đoạn hoàn hảo)

   +O(n) :  Trường hợp đặc biệt,khi các phần tử trong mảng đề bằng nhau==> giảm tối thiểu được các phép biến đổi (Phân đoạn cân bằng)

Vd:Khi ta cho dãy đang giảm dần,mà cần sắp xếp tăng dần.

Nếu ta đặt Pivot ở đầu,nó sẽ vào TH xấu nhất :O(n^2)

Nếu ta đặt ở giữa,thì nó sẽ trung bình : O(nlog(n))

⇒Nên sử dụng pivot ở giữa

+) Hàm sort là sử dụng chia để trị,chọn Pivot ngẫu nhiên,nó sẽ luôn đảm bảo ĐPT luôn ở O(nlog(n))

VD:Pivot ở cuối:

[Mr3xS5 - Online C++ Compiler & Debugging Tool - Ideone.com](https://www.ideone.com/Mr3xS5)

VD cho quá trình phân đoạn:



Binary Search

1.KN: Là 1 thuật toán tìm kiếm phần tử trên 1 dãy đã sắp xếp

2.Các bước:

  +Sắp xếp

  +Lấy phần tử ở giữa so sánh với X.(So sánh giữa vì ta muốn chia dãy làm đôi.Nếu đặt pivot ở đầu thì ta có thể sẽ phải duyệt hết cả dãy => O(n) VD: X=1,dãy:2234567 ,p=7⇒ nó sẽ duyệt lần lượt từ 7 về 2  )

Nếu không bằng nhau,ta bỏ 1 nửa mảng và xét tương tự với nửa mảng còn lại.

  +Nếu tìm được X ++>YES

  +Đến khi kết thúc mà chưa tìm được X thì ==>NO.

-Độ phức tạp :O(log(n))

-Vì sao lại là nhị phân mà ko phải chia 3,4,5......

  +Cài đặt đơn giản,ko phải tạo nhiều biến

  +Đỡ phức tạp

-Lúc nào bắt buộc phải tìm kiếm tam phân

  +Khi cần tìm cực trị của 1 hàm f(x)

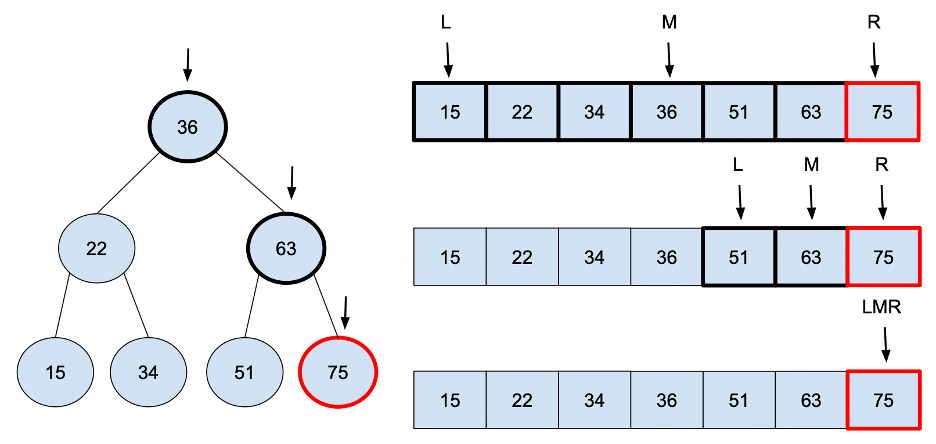
3.Hàm có sẵn trong C++.

binary\_search(a,a+n,x) : chỉ trả về x có xuất hiện hay ko,ko cho biết vị trí của x

Code:

<https://ideone.com/OoMrow>

-Mô tả:



Bài tập.

Bài A:

B1-Tạo hàm quicksort

B2-Xuất ra phần tử lớn thứ K.

BaiH:

B1:Tạo hàm binary-search,và tìm kiếm cây có chiều cao K