**1) Độ phức tạp thuật toán là gì?**

Ví dụ cho 2 câu lệnh:

a)

for i in range(0,n):

s+=a[i]

DPT O(n)

b)

for i in range(0,n):

for j in range(0,m):

s+=a[i][j]

s-=b[i][j]

DPT O(2\*M\*N)

O(n)+ O(2\*M\*N) = O(M\*N)

Question:

Câu lệnh nào nhanh hơn? Nhanh như thế nào?

=> Độ phức thuật toán hiểu đơn giản là số bước thực hiện câu lệnh ký hiệu là O

1 số qui ước :

O(k\*n) = O(n)

O(Log2(N)) = O(LogN)

**2) Một số thuật toán sắp xếp**

#### Insertion sort (O(n^2))

* Selection sort (O(n^2))
* Bubble sort (O(n^2))
* Counting sort (O(N))
* Merge sort
* Quicksort
* Heapsort

Minh họa :

https://sites.google.com/a/sfusd.edu/cs-in-sf-mycs-for-teachers/unit5/5-4-sorting-algorithms-selection-and-bubble-sort

Bài tập:

1) Cho danh sách học sinh lớp A biết chiều cao và cân nặng của n bạn học sinh, sắp xếp danh sách học sinh theo thứ tự tăng dần chiều cao, trong trường hợp chiều cao bằng nhau thì sắp xếp theo cân nặng

2) Cho mảng N phần tử (N rất lớn , mỗi phần tử nằm trong khoảng [0,10^8]

sắp xếp mảng N tăng dần

3) Cho dữ liệu vào là luồng stream gồm các phần tử là số , tại thời điểm bất kì cần xuất ra dãy hiện tại được sắp xếp tăng dần

**3) Thuật toán tìm kiếm**

* Tìm kiếm duyệt thông thường
* Tìm kiếm bằng đánh dấu
* Tìm kiếm nhị phân

Bài tập:

1. Cho dãy số nguyên bất kì : hỏi k có trong dãy số hay không
2. Cho dãy A sắp xếp tăng dần , kiểm tra X có nằm trong dãy hay không
3. Nâng cao : Có N cây gỗ, có chiều cao lần lượt là A[1],A[2],..,A[n]. Bạn cần lấy một lượng gỗ độ cao tối thiểu là M bằng cách chặt từ N cây theo cách như sau: chặt tất cả những phần thừa của các cây có độ cao lớn hơn H. Hãy tìm giá trị H lớn nhất để bạn có thể lấy được lượng gỗ tối thiểu là M.