**Data structure and algorithm**

1. To present – Data structure and algorithm

Tự nghiệm về thuật toán : algorithm là giải quyết một vấn đề với nhiều trường hợp khác nhau bằng một khối thủ tục từng bước 1..

1. Sort - at place - not in place

* Tại chổ thì một số không gian để so sánh và lưu trữ tạm thời một số phần tử dữ
* Và ngược lại

1. Sort – stability – no stability

* ổn định thì không làm thay đổi số vị trí của các phần tử giống nhau
* không ổn định thì thay đổi vị trí của các phần tử giống nhau

1. Sort – Adaptation – no Adaptation

* Thích ứng thì tận dụng được các phần tử đã được sắp xếp trong một mảng
* Không thích ứng thì ngược lại

1. Chú ý quan trọng

* Sort - increase
* Sort – decrease
* Sort - no increase là giống như [ 1 ,3 ,3 ,5,6]
* Sort – no decrease [1, 3, 3, 6, 8, 9

**II – algorithm Selection.**  Ο(n2)

* Compare - two array ->> left half and right half

‘’ nữa trái đã được sắp xếp và **default**  ban đầu không chứa phần từ nào.

And nữa phải chứa đầy đủ các phần từ của mảng ”

Step 1: seach element Mix đưa vào bên left of aray tức là element[0]

Step 2 : compare element [1] với all element in aray

* If nếu elemnet đó nhỏ nhất thì giữ nguyên vị trí
* If phần tử khác trong mảng thì đổi vị trí của 2 phần từ đó
* Phần cedo
* For 1 chạy từ 0 đến length – 1 hoặc length
* Lấy một biến mix để tìm vị trí của phần từ nhỏ nhất
* For 2 bất đầu từ 1 vì element 1 compare với phần tử 2
* If nếu biến mix khác biến I thì chắc chắn đã có một vị trí mix khác nên ta lần lượt swap

Trường hợp sử dụng : Không phát hiện được danh sách đã xếp sắp

**II – algorithm Insert.**  Ο(n2)

1 . take phần tử thứ nhất so với phâng tử thứ 2.

While

**II – Linked list .**  danh sách liên kết

* Linked list in java is a class AbstractSequentialList ( danh sách tuần tự) queue interface in celection framwork

1. Poínt needed memorize

* Class linked in java có thể chứa các phần tử trung lặp
* Duy trì thứ tự của element save in
* Không đồng bộ
* Linked in java có thể được như list, tack, queue

1. Declare

public class LinkedList<E> extends AbstractSequentialList<E>

kéo dài

        implements List<E>, Deque<E>, Cloneable, java.io.Serializable

* Khỏi tạo một linked list

LinkedList<String> list = new LinkedList<String>(); // generic - kiểu mới

* **Have two way**

LinkedList() 🡪 tạo list space

LinkedList(Collection c) 🡪 tạo list các element đã chọn

* Các methor of linked list
* boolean add(Object o) - nối phần tử chỉ định vào cuối list
* void add(int index, Object element) chèn các phần từ theo índex chỉ định
* void addFirst(Object o) – chèn phần tử chỉ dịnh vào list
* void addLast(Object o) – chèn phần từ chỉ định vào cuối list
* int size() – trả về số lượng phần tử trong list
* boolean contains(Object o) – return true if có chứa element
* boolean remove(Object o) – xoá phần tử chỉ định first in list
* Object getFirst() – return element first in list
* Object getLast() return element last in list
* int indexOf(Object o) – return indexing of elemnet seaching or return -1
* int lastIndexOf(Object o) return indexing of phân từ xuất hiện last return -1
* boolean contains(element) - return true if seach thấy element và ngược