# Lý thuyết:

Bài 1: slide 43, 50, 63, 71

Bài 2: pointers vs references

Bài 3: instance vs object

# Thuật toán:

## Dạng 1: Kiểm tra\_Kết luận;

(1) Đưa ra giả thiết;

(2) Kiểm tra giả thiết - nếu thấy sai: ///==> kết luận và dừng việc kiểm tra tiếp

Ví dụ: Nhập vào số nguyên n => ngto hay ko phải là nguyên tố

(1) boolean ktra=true; // gia thiết n là số nguyen to

(2)

for (int i=2; i<=Math.sqrt(n);i++) {

if (n%i==0) {

ktra=false; //

break;

}

} // Khi nào thì bỏ {} được?

if (ktra)

System.out.println(n +" là số nt");

else

System.out.println(n +" ko là số nt");

Bài tập áp dụng (cần làm với nhiều loại vòng lặp khác nhau):

(1) Mảng int [] A={5, 7, 9, 10,16} tạo thành dãy tăng dần hay ko? ==>yes/No

(2)Mảng int [] A={5, -1, 9, -10,16} tạo thành dãy đan dấu hay ko? ==>yes/No

(3) Mảng int [] A={5, 1, 9, 10, 16} ////ktra xem có phải tất cả các phần tử của A đều là số nguyên tố hay ko?

## 2. Dạng 2: Kiểm tra\_Cập nhất giả thiết

### Công thức :

1. Giả thiết;
2. Kiểm tra giả thiết- Nếu giải thiết sai: cập nhật lại giả thiết.

Ví dụ: tìm giá trị LN trong một mảng

int [] A={12, 3, 34, 43, 434, 38, 55 }; // In ra giá trị LN (NN) //

1. int max=A[0]; // Giả thiết A[0] là lớn nhất; // max=0;

for (int i=1; i<A.length; i++) {

if (A[i]>max) max=A[i];

}

System.out.println("Max ="+ max);

### Các bài toán tượng tự:

(1) Cho mảng int [] A={*5, 7*, **1, 9, 10, 16**, 4, 6, 8, 3}. // Hãy xác định độ dài và in dãy con tăng dần dài nhất //

(2) Cho mảng int [] A={5, **1, 9, 10,16**, 4,6, 8, 3}. // Hãy xác định và in dãy con tăng dần dài nhất có tổng các phần tử lớn hơn 20 //

(3) Cho mảng int [] A={5, 1, 9, **10, 16, 4,6, 8**, 3}. // Hãy xác định và in dãy con dài có các số chẵn liên tiếp dài nhất

## 3. Dạng 3: Mở rộng của Dạng 2

### Công thức:

(1) Giả thiết;

(2) Kiểm tra giả thiết: // Giả thiết đúng ==> tăng đếm // Giả thiết sai ==> Cập nhật lại giả thiết; Reset giá trị đếm về 1

Ví dụ: Hãy xác định giá LN và in ra số lần xuất hiện cùng các ví trị xuất hiện của nó

int [] A={12, 3, 434, 12, 43, 434, 38,55 };

int [] pos = new int[A.length];

(1) int max=A[0]; int cnt=1; // Giả thiết A[0] LN và xuất hiện 1 lần

(2) pos[cnt-1]=0;

for (int i=1; i<A.length;i++) {

if (max==A[i]) { cnt++; pos[cnt-1]=i;

}

else if (max<A[i]) { max=A[i]; cnt=1; pos[cnt-1]=i;

}

}

System.out.println("Max = "+ max);

System.out.println("So lần xuất hiện: "+ cnt);

System.out.println("Xuất hiện ở các vị trí sau: ");

for (int i=0;i<cnt;i++) System.out.print(pos[i]+" ");

/\* Thay đổi đoạn in kết quả theo yêu cầu sau:

(1) Nếu chỉ xuất hiện 1 lần: Max = ... ở vị trí: ....

(2) Nếu xuất hiện nhiều lần:

Max = ... Số lần: ....

Ở các vị trí: 3, 5, 6, ...10. \*/

### Bài tập áp dụng:

(1) Cho mảng int [] A={5, 3, 7, 8, 4, 6, 8, 3}. // Hãy xác định và in tất cả các dãy con tăng dần dài nhất cùng các vị trí bắt đầu xuất hiện của nó

(2) Cho mảng int [] A={5, 1, 9, 10,16, 4,6, 8, 3}. // Hãy xác định và in tất cả các dãy con tăng dần dài nhất có tổng các phần tử lớn hơn 100 cùng các vị trí bắt đầu xuất hiện của nó

(3) Cho mảng int [] A={5, 1, 9, 10,16, 4,6, 8, 3}. // Hãy xác định và in ra tất cả các dãy con dài nhất gồm các phần tử là các số chẵn.