## **Bài tập ctdlgt-bt09**

### **Bài 1**

Lấy mã nguồn chương trình và dữ liệu về các cài đặt (implementation) Symbol Tables (không sắp xếp - SequentialSearchST.java và có sắp xếp - ST.java) tại [link mã nguồn môn học](http://algs4.cs.princeton.edu/code/). Chạy lại chương trình với dữ liệu test.

**Chú ý**: Thư viện [algs4.jar](http://algs4.cs.princeton.edu/code/algs4.jar)

### **Bài 2**

[Java Map](https://www.hackerrank.com/challenges/phone-book)

Tóm tắt: Trước tiên, nhập vào danh sách tên người cùng số điện thoại. Sau đó, với mỗi câu hỏi có tên người, in ra số điện thoại tương ứng; Nếu không có người nào có tên đó, thì in ra "Not found".

Gợi ý: Sử dụng Map (tức bảng ký hiệu) có trong Java. Các mã sau là ví dụ về Map:

// Khai báo 1 Map với khoá String và giá trị Integer  
 Map people = new HashMap();  
 //...  
 // Thêm người có tên với số điện thoại vào map  
 people.put(name, phone);  
 //...  
 // Kiểm tra nếu có người trong map thì in ra tên và điện thoại, nếu không in ra "Not found"  
 if (people.containsKey(s)) System.out.println(s + "=" + people.get(s));  
 else System.out.println("Not found");

Link thêm về [Java - The HashMap Class](https://www.tutorialspoint.com/java/java_hashmap_class.htm).

### **Bài 3**

[Java HashSet](https://www.hackerrank.com/challenges/java-hashset)

Tóm tắt: Nhập vào một cặp 2 chuỗi. Sau mỗi lần nhập, in ra số lượng các cặp duy nhất (không kể bị lặp lại) đã được nhập.

Gợi ý: Nối 2 phần trong mỗi cặp thành 1 chuỗi (string) với dấu trắng ở giữa. Như tên bài, sử dụng HashSet để lưu tập dữ liệu (hay có thể dùng Map, nhưng hơi thừa). Và như vậy, sau mỗi lần nhập, chỉ cần in ra kích cỡ của tập đang có.

Link thêm về [Java - The HashSet Class](https://www.tutorialspoint.com/java/java_hashset_class.htm).

### **Bài 4**

[Tree: Inorder Traversal](https://www.hackerrank.com/challenges/tree-inorder-traversal)

Tóm tắt: Viết hàm Duyệt cây theo thứ tự trong.

Gợi ý: Sử dụng đệ qui

### **Bài 5**

[Tree: Preorder Traversal](https://www.hackerrank.com/challenges/tree-preorder-traversal)

Tóm tắt: Viết hàm Duyệt cây theo thứ tự trước.

Gợi ý: Cách 1: Sử dụng đệ qui; Cách 2: Sử dụng stack (ngăn xếp).

### **Bài 6**

[Tree: Level Order Traversal](https://www.hackerrank.com/challenges/tree-level-order-traversal)

Tóm tắt: Viết hàm Duyệt cây theo thứ tự từng hàng một (từ trên xuống dưới).

Gợi ý: Sử dụng hàng đợi (queue) để lưu các đỉnh cần duyệt (lần lượt cha, rồi sau đó là con trái, con phải ...)

### **Bài 7**

Làm lại với **yêu cầu mới** bài [Closest Numbers](https://www.hackerrank.com/challenges/closest-numbers)

Tóm tắt: Tìm và in ra 2 số có chênh lệch nhỏ nhất.

Yêu cầu: **Không** sử dụng hàm sắp xếp (sort) có sẵn của Java. Thay vào đó, **tự viết** phần sắp xếp theo thuật toán **sắp xếp Hoà trộn** (Merge Sort).