## Bài tập ctdlgt-bt11

### Bài 1

Lấy mã nguồn chương trình và dữ liệu về các cài đặt (implementation) khác nhau của Bảng băm (Hash Tables) theo 'separate chaining' và 'linear probing' tại [link mã nguồn môn học (phần searching)](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

a. Thực hiện các thao tác sau (vẽ/chạy bằng tay) với cây thứ tự bộ phận (cây max-heap):

- Thêm lần lượt vào cây: 1, 3, 10, 9, 6, 7, 4, 5, 2, 8

- Lấy và xoá giá trị lớn nhất (ở gốc) 4 lần liên tiếp.

b. Thực hiện các thao tác sau đây với cây tìm kiếm nhị phân:

- Thêm lần lượt vào cây: 1, 3, 10, 9, 6, 7, 4, 5, 2, 8

- Xoá ra khỏi cây: 3, 5, 7, 9

c. Thực hiện tương tự như ở câu (b) nhưng với cây tìm kiếm cân bằng AVL.

### Bài 3

a .Chạy minh hoạ (vẽ/chạy bằng tay) bảng băm (hash table) dạng 'linear probing' (thăm dò tuyến tính) 'open addressing' (đánh địa chỉ mở) với:

- Mảng chứa các khoá có độ dài 17 phần tử.

- Khoá là các số nguyên.

- Chèn các khoá vào lần lượt là 3, 13, 6, 7, 23, 16, 33

- Tìm kiếm các khoá 6, 33, 40

b. Chạy minh hoạ (vẽ/chạy bằng tay) bảng băm (hash table) dạng 'separate chaining' có độ dài 17, với các thao tác:

- Chèn các khoá vào lần lượt là 3, 13, 6, 7, 23, 16, 33

- Tìm kiếm các khoá 6, 33, 40

### Bài 4

Làm lại với **yêu cầu mới** bài [Pairs](https://www.hackerrank.com/challenges/pairs)

Tóm tắt: Đếm số các cặp số nguyên có chênh lệch là *K*.

**Yêu cầu**: Trong thuật toán có sử dụng bảng băm (Hash Tables/Sets)

Gợi ý: Trong slide bài giảng "Phân tích thuật toán" có thuật toán cho 3-sum. Bài này có thể xem như biến dạng của 2-sum. Như thế có thể giải theo cách: với mỗi số *n*, kiểm tra có số nào thoả mãn điều kiện là chênh với số *n* một giá trị là *K*.

### Bài 5

Làm với **nhiều cách** bài [Mising Numbers](https://www.hackerrank.com/challenges/missing-numbers)

Tóm tắt: Cho 2 danh sách A và B. (Chênh lệch giữa số lớn nhất và nhỏ nhất của B là không quá 100.) Tìm và in ra các số có trong B nhưng không có trong A theo trật tự tăng dần.

**Yêu cầu**: Làm bài này bằng 3 cách khác nhau. Phân tích độ phức tạp tính toán của mỗi phương pháp.

Gợi ý:

- Cách 1: Sử dụng bảng băm (Hash Tables) để đếm và tìm; sau đó sắp xếp để in ra.

- Cách 2: Chỉ sử dụng sắp xếp (Sorting) khi tìm và in ra.

- Cách 3: Sử dụng cách làm trong bài Sắp xếp đếm (Counting Sort)

### Bài 6

Làm lại với **yêu cầu mới** bài [Find the Running Median](https://www.hackerrank.com/challenges/find-the-running-median)

Tóm tắt: Đọc lần lượt vào **N** số nguyên. Sau mỗi lần đọc vào số thứ **i**, in ra giá trị trung vị của **i** số đã nhập vào.

Với dãy có (2k + 1) số, thì giá trị trung vị bằng phần tử đứng giữa (có k số nhỏ hơn (hoặc bằng) nó, và đồng thời có k số lớn hơn (hoặc bằng) nó). Với dãy có (2k) phần tử, giá trị trung vị bằng trung bình cộng hai phần tử đứng giữa.

**Yêu cầu**: Trong thuật toán có sử dụng cây tìm kiếm nhị phân (Binary Search Tree)

Gợi ý:

* Sử dụng TreeSet trong Java, với phương thức higher() và lower().
* Để giải quyết vấn đề các số nguyên có thể bằng nhau, mỗi phần tử trong TreeSet có thể là một lớp gồm hai thuộc tính: giá trị số nguyên và thứ tự số nguyên đó.