

Địa chỉ email gửi bài: vingovan@admicro.vn

Gửi vào T5 - trước 11h hàng tuần

Nội dung gửi bao gồm:

- Kết quả đạt đc (doc + code có thể cho lên gitlab hoặc github)
- Các khó khăn, câu hỏi nếu có

Tuần 1:

Bài 1: Cho 2 mảng đã được sắp xếp. Tìm giao của hai mảng với độ phức tạp là $O(m \log(n))$ với m và n tương ứng là độ dài của 2 mảng

Kiến thức cần tìm hiểu:

- Độ phức tạp thuật toán
- Tìm kiếm nhị phân

Bài 2: Một url là url thỏa mãn các yếu tố

(ví dụ 1 url hợp lệ: <https://tiki.vn/dien-thoai-may-tinh-bang/c1789?src=mega-menu>):

- Bắt đầu bằng http hoặc https
- Có thể chứa www hoặc không
- Tên miền của url (ví dụ tiki) chỉ chứa các ký tự la tinh hoa/thường từ a-z và chữ số.
- Phần domain extension (ví dụ .vn) phải chứa dấu chấm ở đầu, thêm sau là các ký tự in thường có độ dài từ 2 đến 6 ký tự.
- Phần path (ví dụ /dien-thoai-may-tinh-bang/c1789?src=mega-menu) không được chứa dấu cách.

Viết đoạn regex để kiểm tra xem một url bất kỳ có là hợp lệ hay không.

Ôn và learn kiến thức về Java:

OOP core:

<http://www.w3resource.com/java-tutorial/java-object-oriented-programming.php>

JVM: Heapsize, dọn rác

Concurrency:

<http://winterbe.com/posts/2015/04/07/java8-concurrency-tutorial-thread-executor-examples/>

(cái này nó viết bằng syntax của java 8)

Collection(s)

(Overview) <http://cs.lmu.edu/~ray/notes/collections/>

(compare) <http://www.codejava.net/java-core/collections/java-collections-framework-summary-table>

(performance) <http://infotechgems.blogspot.com/2011/11/java-collections-performance-time.html>

String processing:

<http://zetcode.com/lang/java/strings/>

<https://www.dotnetperls.com/format-java>

I/O:

(phần rút gọn) <http://www.studytonight.com/java/java-io-stream.php>

(phần đầy đủ) http://www.ntu.edu.sg/home/ehchua/programming/java/j5b_io.html

(phần đầy đủ khá phức tạp, và nhiều phần, em đọc lướt thôi nhé)

Exception

<http://www.journaldev.com/1696/java-exception-handling-tutorial-with-examples-and-best-practices>

SQL & JDBC

Sử dụng được các lệnh Insert, select, update, delete. (Tài liệu tìm trên mạng)

JDBC:

<http://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples>

Tuần 2:

Bài 1: Sắp xếp 1 stack sử dụng 1 stack

Bài 2:

Cho hai lớp Country và City có các thuộc tính (như bên dưới). Một quốc gia có 1 thủ đô và có thể có 1 hoặc nhiều thành phố. Một thành phố luôn trực thuộc chỉ 1 quốc gia.

- Country:

- + String code (mã quốc gia)
- + String name
- + String continent (mã lục địa)
- + double surfaceArea (diện tích bề mặt)
- + int population (dân số)
- + double gnp (Gross National Product)
- + int capital (mã thành phố là thủ đô của đất nước này)

- City:

- + int id
- + String name
- + int population (dân số của thành phố)

Tự tạo dữ liệu dựa trên bảng trên

sau đó sử dụng lambda và stream api trong java 8 để thực hiện các yêu cầu sau:

- 1.1 Tìm thành phố đông dân nhất của mỗi quốc gia.
- 1.2 Tìm thành phố đông dân nhất của mỗi lục địa.
- 1.3 Tìm thành phố là thủ đô, đông dân nhất.
- 1.4 Tìm thành phố là thủ đô, đông dân nhất của mỗi lục địa.
- 1.5 Sắp xếp các quốc gia theo số lượng thành phố giảm dần.
- 1.6 Sắp xếp các quốc gia theo mật độ dân số theo thứ tự giảm dần bỏ qua các quốc gia có dân số bằng không.

Bài 3: Tìm hiểu spark, hadoop

Tuần 3:

Bài 1:

Tìm vị trí đầu tiên và cuối cùng của một thành phần trong mảng đã sắp xếp

input: một mảng đã được sắp xếp tăng dần và một số

output: vị trí đầu tiên và cuối cùng của số đã cho

nếu không tồn tại thì trả về -1

Ví dụ:

Input: nums = [5,6,7,8,8,15], target = 8

Output: [3,4]

Yêu cầu độ phức tạp: $O(\log n)$

Bài 2:

- Tìm hiểu Kafka (mô tả dạng 5w1h)
- Giả lập việc đẩy dữ liệu vào kafka và dùng spark streaming cho việc tính toán

Tuần 4:

Bài 1: Tìm hiểu về Prefix Sum và ứng dụng trong bài toán counting sort

Bài 2: Tìm hiểu về một số design pattern. Mô tả và implement lại chúng

- SINGLETON
- FACTORY
- Chain of Responsibility
- Adapter Pattern

Phần seminar:

Bài 3: Tìm hiểu về microservice (chuẩn bị slide cho trình bày seminar, implement)

Tìm hiểu CQRS và Event Sourcing

Bài 4: Tìm hiểu hệ thống data lake (chuẩn bị slide, demo)

<https://delta.io/>

