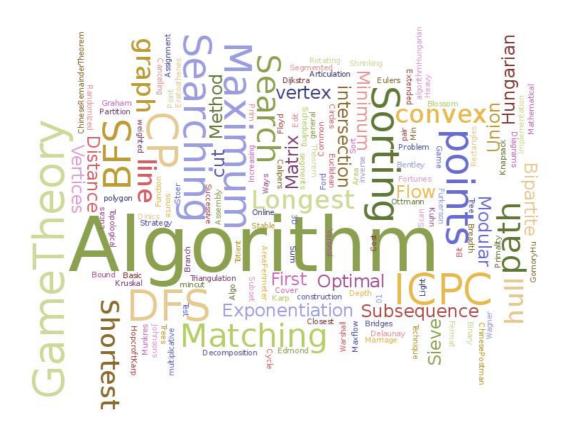
Data structures and algorithms



- Data structures and algorithms
 - o 1. Giới thiệu
 - 1.1 Mục tiêu
 - 1.2 Hình thức
 - 1.3 Báo cáo
 - o 2 Khái niệm
 - 2.1 Cấu trúc dữ liệu
 - 2.2 Design Pattern
 - 2.3 Nguyên tắc lập trình
 - o 3. Bài tập
 - 3.1 Predictive text
 - 3.2 Hash Tables
 - 3.3 Tính thời gian xử lý khiếu nại
 - o 4. Tham khảo

1. Giới thiệu

1.1 Mục tiêu

- Cung cấp kiến thức về lập trình.
- Củng cố nền tảng về cấu trúc dữ liệu và giải thuật.
- Làm quen với các design pattern phổ biến.
- Nắm vững nguyên lý lập trình cơ bản.

• Làm quen với Java (dùng Intellij và tạo project Maven)

1.2 Hình thức

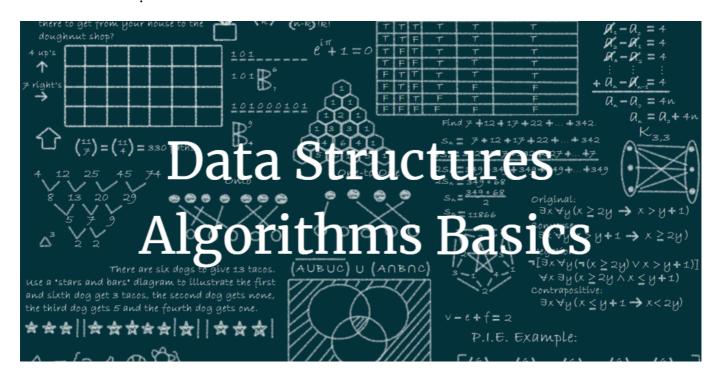
- Keyword: Định hướng cách thức nghiên cứu lý thuyết, viết report.
- Bài tập: Định hướng cách thực hành, sử dụng giải thuật cơ bản.
- Đồ án: Kết hợp giữa lý thuyết và thực hành để xây dựng một sản phẩm hoàn chỉnh.
- Module này thang điểm đánh giá là 10.
- Mức điểm đánh giá:
 - <= 6: Chưa đạt, cần cải thiện kiến thức.</p>
 - >= 7: Nắm được kiến thức và ứng dụng được.

1.3 Báo cáo

- Tạo repo cá nhân chứa note và code liên quan.
- Thời gian: 2 tuần

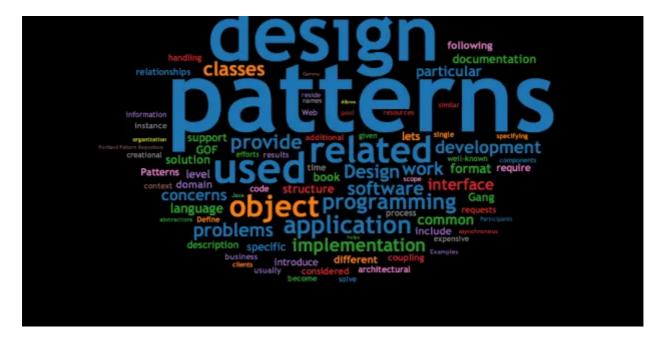
2 Khái niệm

2.1 Cấu trúc dữ liệu



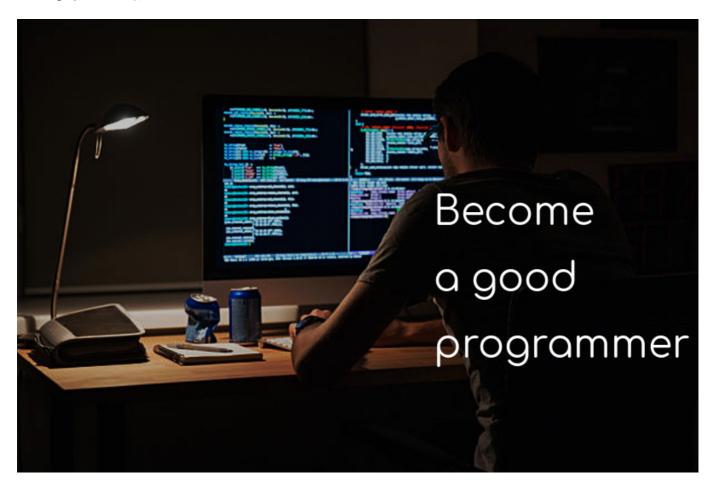
- Probabilistic data structures có tính chất thế nào? Nêu cách thức hoạt động của Bloom Filters, Cuckoo Filters, Count Min Sketch, HyperLogLog).
- Trie là gì? Ứng dụng như thế nào?

2.2 Design Pattern



- Nêu 5 design pattern hay dùng (Dependency Injection, Factory, Singleton, Builder, ...)
- Code minh họa cho từng design pattern.

2.3 Nguyên tắc lập trình



Tìm hiểu các nguyên tắc sau:

- SOLID
- DRY
- KISS
- YAGNI

• Do the simplest thing that could possibly work

Clean code là gì? Nêu ít nhất 5 cách để clean code?

3. Bài tập

3.1 Predictive text

Cho dataset Blog Authorship Corpus.

Làm chương trình Java cung cấp 2 tính năng:

- Kiểm tra 1 từ có nằm trong dataset không?
- Gợi ý những từ giống với từ input mà có trong dataset (gần giống như cách Google instant search gợi ý).
 Sự gần giống giữa các từ do em tự định nghĩa nhưng phải hợp lý.

Yêu cầu:

- Phải hiện thực ít nhất 2 cách với tính năng kiểm tra từ tồn tại.
- Tìm cách tối ưu chương trình với các cấu trúc dữ liệu mà em đã học được ở trên.
- Benchmark cho phần kiểm tra từ.
- Sử dụng ít nhất 2 design pattern.
- Không được import lib có sẵn(được copy source hiện thực nhưng sẽ có hỏi về cách hoạt động).
- Phần gợi ý phải gợi ý ít nhất 5 từ gần giống với input đầu vào.
- Tự định nghĩa cách hiện output.
- Nhớ viết chương trình như một Good Programmer

Tip: Nên định nghĩa 1 interface là Dictionary với method public boolean contains (String word) để implement lại.

3.2 Hash Tables

- Tham khảo repo sau.
- Viết lại hoàn toàn bằng Java một hash table tương tự.
- Hiện thực ít nhất 3 cách giải quyết đụng độ.

3.3 Tính thời gian xử lý khiếu nại

Hiện tại, bộ phần CS(Customer Service) sẽ nhận các khiếu nại từ người dùng và trả lời các khiếu nại đó. Để người dùng không phải chờ câu trả lời quá lâu, chúng ta phải đảm bảo người dùng sẽ nhận câu trả lời trong thời gian tối đa là 8 tiếng làm việc. Vì vậy, cần một chương trình tính thời gian từ lúc nhận khiếu nại đến khi khiếu nai được giải quyết để biết bộ phận CS làm việc có hiệu quả.

Giờ làm việc được tính từ 8h30 tới 12h với buổi sáng và 13h30 đến 18h với buổi chiều. CS sẽ làm nguyên ngày từ thứ 2 đến thứ 6, riêng thứ 7 sẽ chỉ làm buổi sáng.

Trong folder ticketSLA là 1 project java đã được init sẵn. Bạn hãy hiện thực hàm calculate ở class SlaServiceImpl. Sau đó, các bạn thêm testcase trong SlaServiceTest để kiểm tra tính đúng sai của hàm đã viết.

Lưu ý:

- Đây là 1 bài toán thực tế nên không có bất kì giới hạn nào.
- Nếu có bất kì thắc mắc nào, vui lòng liên hệ thinhda.
- Benchmark cho hàm calculate.

4. Tham khảo

- Principles of Good Programming
- Programming Principles
- Clean code
- Series 4 bài về Probabilistic data structures của Andrii Gakhov.
- One secret to becoming a great software engineer: read code
- Awesome Algo.
- Cracking the Coding Interview: 150 Programming Questions and Solutions.
- Data Structures and Algorithms in Java (2nd Edition).
- jupyter