ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH KHOA KHMT

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN LỚP CS112.M11.KHTN

BÁO CÁO CUỐI KÌ

Nhóm 5:
- Nguyễn Quốc Huy Hoàng – 20520051
- Lê Nguyễn Khánh Nam – 20520073

Yêu cầu: Viết bài thu hoạch về những gì nhận được từ môn học.

Qua môn CS112 (Phân tích và thiết kế thuật toán) thì tụi em đã học tập và tìm hiểu được thêm rất nhiều kiến thức mới mẻ về môn học cũng như là một số các kĩ năng mềm, đồng thời có thể ôn tập lại nhiều kiến thức đã được học từ trước.

A. Về kiến thức nền tảng của môn học

Ở môn học này tụi em được tự do thảo luận và rèn luyện về rất nhiều các chủ đề khác nhau, điều này khiến cho tiết học về thuật toán trở nên năng động hơn, mới mẻ hơn, có cảm hứng hơn. Các môn học về lập trình hay là thuật toán trước mà tụi em đã được học như là Nhập môn lập trình, Cấu trúc dữ liệu và giải thuật, Lập trình hướng đối tượng thì tụi em được học tập bằng ngôn ngữ đó là C++, nhưng môn học này tụi em được tiếp xúc nhiều với một ngôn ngữ khác đó là Python. Được học tập với một ngôn ngữ lập trình mới thì tụi em ban đầu cũng gặp nhiều khó khăn, nhưng sau đó đã cố gắng tìm hiểu và củng cố kiến thức của mình về Python. Tụi em đã học được khá nhiều điều mới mẻ về ngôn ngữ này như là viết code ngắn gọn, nhiều thư viện hỗ trợ, …

Trong quá trình học tập thì tụi em đã cùng nhau tìm hiểu và thảo luận về 11 topic chính, cụ thể là:

1. Computational Thinking

Đây là chủ đề đầu tiên mà tụi em được thảo luận. Biết được cách suy nghĩ và giải quyết bài toán theo bằng Computational thinking. Hiểu và áp đụng được vào bài toán các khái niệm về Abstraction, Pattern recognition, Decomposition, Algorithm design, Testing, Evaluation.

2. Phân tích độ phức tạp thuật toán

Ở các cuộc thi về tin học thì phần tính độ phức tạp rất là quan trọng, nên chắc ai đã từng tham gia các cuộc thi thì cũng đã học qua rồi, và team em cũng vậy, nên khi nghe về tên của chủ đề thì tưởng đây là chỉ chủ đề nhàm chán. Nhưng ở 2 buổi thảo luận của chủ đề này thì tụi em vẫn học và biết thêm nhiều điều mới mẻ khác, như là biết cách tính ĐPT chuẩn chỉnh hơn, dùng định lý Master để tính độ phức tạp. Trước kia tính ĐPT của thuật toán thì tụi em thường ước lượng, nhưng qua buổi thảo luận thì tụi em cảm thấy mình quan sát và tính toán được độ phức tạp của thuật toán chuẩn hơn. Qua buổi học này thì tụi em cũng tham gia giải các bài tập trên codeforces, và cũng đã chú ý hơn về việc tính toán độ phức tạp của các thuật toán của mình trước khi tiến hành code.

3. Kiểm tra tính đúng đắn & hiệu năng của chương trình bằng bộ test

Qua một số các bài tập về sinh test mà team chủ trì thảo luận đưa ra, thì team em cũng biết được nhiều hơn về cách sinh test cho các dạng bài toán, như là sinh chuỗi, sinh cây, sinh đồ thị, … cũng như là về tư duy sinh test, sinh test làm sao mà vẫn đảm bảo về hiệu năng cũng như là độ chính xác. Vì là từ chủ đề này trở đi là các chủ đề sau phải sinh test để chấm bài tập về nhà, nên team của em cũng đã tự ôn luyện và học được thêm nhiều điều mới, như là nhiều các thư viện sinh đồ thị bằng Python, rồi các thao tác với file trong Python.

4. Phương pháp thiết kế thuật toán: Completed search - Brute force

Đây là một chủ đề cơ bản, dễ hiểu, dễ cài đặt, và team của tụi em cũng đã có tìm hiểu và làm nhiều bài tập từ trước. Nhưng ở chủ đề này tụi em vẫn biết thêm được một số kiến thức từ trong buổi học và tự tìm hiểu sau đó như là các ưu nhược điểm của Brute Force, sự phổ biến rộng rãi của Brute Forces, …

5. Phương pháp thiết kế thuật toán: Divide and Conquer

Biết thêm được về thuật toán Karatsuba áp dụng chia để trị. Đồng thời qua các bài tập trong buổi thảo luận thì tụi em cũng ôn luyện lại được các kiến thức mà trước kia đã học. Hiểu rõ hơn về ưu, nhược của thuật toán chia để trị, có thể nhận dạng được bài toán nào có thể sử dụng chia để trị một cách tốt hơn.

6. Phương pháp thiết kế thuật toán: Greedy approach

Đây là một thuật toán khá quen thuộc, team em đã học từ trước đó. Qua chủ đề này thì tụi em có thể ôn luyện lại được kiến thức cũ. Ngoài bài toán đổi tiền ATM thì còn biết được thêm một số các ứng dụng thực tế của thuật toán tham lam như là xếp lịch, tìm đường đi, nén dữ liệu, …Tụi em cũng đã có tìm hiểu và tự làm bài tập về chủ đề này trên codeforces, vnoi và đã nâng cao khả năng nhận biết một bài toán có thể giải bằng thuật toán Greedy được hay không.

7. Phương pháp thiết kế thuật toán: Completed search – Backtracking

Biết được khi nào thì nên dùng Backtracking, sự khác nhau của Backtracking và Brute force, ưu nhược điểm của Backtracking. Biết được nhiều ứng dụng thú vị của Backtracking như là AI, Network communication, Robotics, Electrical Engineering.

8. Phương pháp thiết kế thuật toán: Completed search - Branch and Bound

Một bài toán có thể có nhiều cận, các cách cài đặt khác nhau thì có thể đặt cận khác nhau. Qua buổi thảo luận về nhánh cận thì team em đã cải thiện được kĩ năng đặt cận tốt hơn, biết được thêm các khái niệm như là lower_bound(X), bestconfig, …, biết được cách đặt cận sao cho hợp lý hơn. Nhận dạng được bài toán nhánh cận tốt hơn, nắm được các ưu điểm, nhược điểm của nhánh cận. Biết được thêm nhiều ứng dụng hơn, không chỉ trong thuật toán đơn thuần mà còn là về các ứng dụng thực tế như là vấn đề cắt giảm cổ phiếu, tính toán phát sinh chủng loài, …Sau khi làm các bài tập về nhà cũng như luyện tập thêm trên mạng thì cả 2 thành viên trong team cảm thấy tự tin hơn về phương pháp này.

9. Phương pháp thiết kế thuật toán: Dynamic programming

Có thể nhận dạng được bài toán quy hoạch động, qua các bài tập thì nâng cao khả năng tìm công thức truy hồi. Ưu nhược điểm của quy hoạch động với các thuật toán đã học trước đó như là Brute force, Greedy, Backtracking, Branch and Bound. Biết được một số ứng dụng của DP như là trong NLP, RL, CV, …

10. Các thuật toán hình học: Geometric Algorithms

Giúp ôn luyện kiến thức về toán hình học, đồng thời có thể biết được cách áp dụng các công thức toán hình học vào bài toán lập trình, như là tính góc, tính diện tích của đa giác, tính khoảng cách giữa các điểm, …Qua các bài tập thì ôn luyện lại được (do team của em đã biết từ trước) một số thuật toán hình học như là Sweep-line, Graham. Biết được thêm về ứng dụng của hình học trong thực tế như là về Machine Learning, đồ họa máy tính, hệ thống thông tin địa lý.

11. Các thuật toán trên đồ thị: Graph Algorithms

Đây là chủ đề mà nhóm tụi em làm chủ trì. Vậy nên đây là chủ đề mà tụi em biết thêm được nhiều điều nhất. Ôn lại kiến thức về đồ thị như là đồ thị là gì, nhận dạng đồ thị, một số các dạng đặc biệt như là đồ thị 2 phía, cây …Biết được rất nhiều ứng dụng của đồ thị trong các thuật toán tin học, cũng như trong thế giới thực. Nắm rõ hơn và chắc kiến thức hơn về 2 thuật toán Dijkstra và MaxFlow. Giúp cải thiện khả năng ra đề, sinh test. Và rất nhiều kĩ năng mềm khác.

Ngoài các buổi thảo luận về chủ đề mới, thì bọn em còn có thêm vài buổi để ôn tập làm về các chủ đề đã học. Việc này giúp cho tụi em nhớ lại và có thể nắm chắc được các kiến thức đã học hơn.

B. Về một số các kĩ năng mềm

Môn này bọn em được tự do thảo luận và học tập lẫn nhau, nên bọn em đã học được thêm rất nhiều kĩ năng mềm khác như là:

- Kĩ năng giao tiếp: Bọn em biết cách đặt câu hỏi cũng như là trả lời câu hỏi lưu loát hơn, đúng trọng tâm và dễ hiểu hơn.
- Tư duy phản biện: Trong buổi thảo luận thì có nhiều ý kiến và cũng có các ý kiến trái nhiều nhau, qua nhiều buổi thảo luận như thế thì tụi em cũng đã trau dồi thêm về khả năng đánh giá và phân tích các luồng thông tin…
- Kĩ năng thuyết trình: Nói lưu loát hơn, biết cách điều phối buổi seminar, đặt câu hỏi và trả lời câu hỏi tốt hơn. Cải thiện và nâng cao kĩ năng làm slide thuyết trình.
- Kĩ năng làm việc nhóm: Phối hợp với nhau tốt hơn, phân chia công việc rõ ràng, công bằng và phù hợp với kĩ năng của mỗi người.
- Kĩ năng tự học: Ngoài làm bài tập trên lớp thì tụi em cũng có thể tự học và nghiên cứu các kiến thức mở rộng về các chủ đề đã học, khả năng tự tìm kiếm thông tin được nâng cao, …