Lesson 1 - Ôn Tập Python

*Created: March 29, 2021*

*Updated: June 02, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Làm quen**
   * Làm quen học viên, xây dựng các mối quan hệ trong lớp học
   * Các học viên trong lớp nhớ tên nhau và nhớ tên giảng viên
   * Giảng viên nhớ tên tất cả các thành viên trong lớp
2. **Định hướng**
   * Giới thiệu về nội dung khóa học.
   * Làm học viên hứng thú với nội dung học.
   * Học viên thu được gì sau khóa học
3. **Kiến thức**
   * Cài đặt môi trường lập trình Jupyter Notebook
   * Học viên ôn tập các kiến thức lập trình Python đã học
4. **Rules lớp học:**

* Chắc chắn phải có rules lớp học, không để xảy ra tình trạng lớp học tự do và không có kỷ luật nhưng phải đảm bảo được tính thoải mái và môi trường cởi mở.
* Một số gợi ý: đi học muộn quá 10p sẽ bị phạt; không làm BTVN sẽ bị phạt; nghỉ học tự do; chơi game trong giờ học,... Có phạt thì phải có thưởng, lưu ý kỹ phần này để đưa ra cách hình thức hợp lý.
* Lưu ý phải có BTVN và học viên phải confirm bài vào một ngày cố định trong tuần. Kèm theo phải chủ động trong việc tự xem video tại nhà. (Để theo lộ trình học mới, học sinh phải có phần commitment cao hơn, giảng viên cũng vậy)
* Giảng viên chú ý nên tạo không khí vui vẻ khi thiết lập RULE. và thường xuyên áp dụng rule ngay khi thiết lập. Khi giới thiệu 1 RULE nên lấy ví dụ cho dễ hình dung:
  + Listen: khi giảng viên nói thì phải lắng nghe để thể hiện sự tôn trọng
  + Hỏi đúng nội dung: không hỏi linh tinh
  + Không làm việc riêng: không chơi game, xem youtube,...

# Chuẩn Bị

**Giảng viên:**

* Xem lại tài liệu bao gồm Syllabus, Lesson Plans, các link reference, các notebook bài học và bài tập.
* Nội dung chính của bài giảng nằm trong Lesson Plans và notebook *Contents/Bài 1. Ôn Tập Python*. Giảng viên dùng notebook này làm material chính khi giảng dạy theo dạng code-along và/hoặc giảng giải code và lý thuyết. Ở cuối notebook có các bài tập để học viên giải và hoàn thành trên lớp. Giảng viên sửa bài cuối giờ, nếu không kịp có thể để học viên tiếp tục làm ở nhà và sửa bài vào buổi làm bài online.
* Sau buổi học, giảng viên giao bài tập về nhà trong notebook *Homeworks/Homework 1. Ôn Tập Python*. Sửa bài ở buổi làm bài online.
* Mỗi tuần sẽ có một buổi học offline để dạy lý thuyết và một buổi làm bài tập online để thực hành, mỗi buổi có thời lượng 2h. Trong buổi online, giảng viên sửa tất cả bài tập đã giao và cho học viên làm bài trong notebook *Labs/Online Practice 1. Ôn Tập Python*. Mỗi bài tập có bộ test case để kiểm tra bài làm của học viên. Giảng viên hướng dẫn để pass tất cả test case và phân tích thuật toán. Hướng dẫn học viên cố gắng giải mà không xem test case.
* *Chú ý:* Giảng viên xóa các bài giải mẫu trước khi gửi các notebook cho học viên, chỉ giữ lại các bài ví dụ. Đối với các notebook *Online Practice*, giữ lại tên hàm và parameter, nội dung hàm thay bằng *pass*.

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Chào hỏi** | * Chơi game bonding đầu buổi học | 10 |  |
| **Giới thiệu khóa học** | * Giảng viên giới thiệu khóa học * Thuật toán là một tập hữu hạn các bước cần thực hiện để giải quyết một vấn đề nào đó. Thuật toán được phát triển theo quá trình phát triển của nhân loại:   + Trước công nguyên: tính tay phép chia 2 số nguyên   + Thế kỷ 20: các thuật toán tìm đường đi ngắn nhất   + Hiện tại: các thuật toán về machine learning/AI   => Học thuật toán là thừa hưởng những kiến thức của nhân loại   * Ứng dụng của thuật toán:   + Tìm kiếm trên Google, tìm đường trên Google Maps   + Thuật toán mã hóa để thanh toán online, là cơ chế hoạt động của tài khoản bitcoin (public-private key)   + AI: xe tự lái, trợ lý ảo Siri * Tại sao nên học khóa học này:   + Xây dựng tư duy thuật toán và giải quyết vấn đề   + Tạo nền tảng để có thể tự tìm hiểu các thuật toán mới và học các mức cao hơn của thuật toán   + Bước đầu xây dựng kiến thức cho các cuộc thi lập trình   + Thuật toán là kiến thức thường được hỏi trong tuyển dụng ngành CNTT | 10 | * [Lịch sử thuật toán](https://en.wikipedia.org/wiki/Algorithm) |
| **Làm quen với Jupyter Notebook**   * Cài đặt Jupyter Notebook * Làm quen với giao diện và cách sử dụng | * Giới thiệu Jupyter Notebook, tại sao nên sử dụng:   + Thuận tiện cho các đoạn code nhỏ   + Thuận tiện cho xử lý data và visualization * Hướng dẫn tải và cài đặt Notebook (chú ý sử dụng Python 3) * Hướng dẫn mở và sử dụng Notebook:   + Cách 1: Mở Anaconda Navigator để chạy Notebook.   + Cách 2: Mở Anaconda Prompt => cd đến thư mục chứa notebook => gõ lệnh “jupyter notebook”   + Hướng dẫn tạo file, tạo cell markdown và code | 20 | * Jupyter Notebook thường được cài đặt chung trong Anaconda. * [Link cài Anaconda](https://www.anaconda.com/products/individual) * Có thể có lỗi khi cài đặt cho từng máy, làm kéo dài thời gian. Có thể bỏ qua đến cuối giờ để xử lý riêng các máy bị lỗi. Cho học viên chạy tạm trên Google Collab |
| **Các thành phần cơ bản của Python** | * Review lại các thành phần cơ bản. * Biến, hằng, kiểu dữ liệu * Phép toán * Input, output * Hàm * Module | 15 | * Nội dung trong notebook *Contents/Bài 1. Ôn Tập Python* |
| **Cấu trúc điều khiển**   * Cấu trúc điều kiện * Cấu trúc lặp * Exceptions | * Cấu trúc điều khiển: if, else, elif * Cấu trúc lặp   + Vòng for: lặp bằng item và index   + Vòng while * Xử lý exception: try except | 20 |  |
| **OOP**   * Class * Object * Attribute * Method * Tính kế thừa | * Class và Object * Attribute:   + Class & instance attribute   + Public & private attribute * Method:   + Constructor   + Public & private method * Tính kế thừa:   + Là khả năng tạo class mới sử dụng các thành phần của class đã có và tiếp tục triển khai chi tiết hơn   + Tính kế thừa được dùng để tận dụng các phần code đã có | 20 |  |
| **Các cấu trúc dữ liệu sẵn có**   * Định nghĩa, ý nghĩa * Có thứ tự: list, tuple, string * Không có thứ tự: set, dictionary | * Cấu trúc dữ liệu là một cách tổ chức cao hơn của những kiểu dữ liệu cơ bản như int, float, boolean,... * Cấu trúc dữ liệu tổ chức và lưu trữ dữ liệu theo các cách khác nhau để thuận tiện cho việc xử lý. Một cấu trúc dữ liệu định nghĩa mối quan hệ giữa các thành phần dữ liệu và các phương thức xử lý trên dữ liệu đó. * Cấu trúc dữ liệu có thứ tự: list, tuple, string * Cấu trúc dữ liệu không thứ tự: set, dictionary | 10 |  |
| **Thực hành**   * Ôn tập giải các bài toán cơ bản trên Notebook | * Giảng viên hướng dẫn làm các bài tập ở cuối notebook * Giảng viên giao bài tập về nhà | 15 | * Bài tập về nhà trong notebook *Homeworks/Homework 1. Ôn Tập Python* |

Lab 1 - Thực Hành Online

*Created: March 29, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức về Python trong Lesson 1

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Giảng viên hướng dẫn làm bài ví dụ, giải thích test cases là gì và đảm bảo các học viên pass tất cả test trong bài ví dụ. * Giảng viên hướng dẫn học viên cố gắng giải mà không xem test cases. * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập trong notebook *Labs/Lab 1. Ôn Tập Python*. Mỗi notebook bài tập có một notebook test cases tương ứng. * Đáp án nằm trong thư mục *Labs/Đáp Án* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 2 - Giới Thiệu về Thuật Toán và Cấu Trúc Dữ Liệu

*Created: April 05, 2021*

*Updated: June 02, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Ý nghĩa của thuật toán và cấu trúc dữ liệu
* Mối quan hệ giữa thuật toán và cấu trúc dữ liệu
* Các cấu trúc dữ liệu sẵn có trong Python

1. **Kỹ năng**

* Tính độ phức tạp của thuật toán
* Cách phân tích một bài toán về lập trình

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. * Có thể chơi Kahoot để tăng tính sôi động. | 15 |  |
| **Giới thiệu thuật toán**   * Thuật toán là gì * Ý nghĩa * Độ phức tạp | * Nhắc lại khái niệm: Thuật toán là một tập hữu hạn các bước cần thực hiện để giải quyết một vấn đề nào đó. * Ý nghĩa thuật toán: Ví dụ với bài toán sắp xếp:   + Bài toán sắp xếp dễ thấy nhất trong chức năng sắp xếp file của Windows (theo thời gian, theo loại file,...) hoặc sắp xếp video trên kênh Youtube (theo thời gian, lượt view,...)   + Giả sử cần sắp xếp 1 tỉ (10^9) số, thuật toán đơn giản nhất: tìm số lớn nhất và đặt lên đầu đến khi hoàn tất, sẽ cần khoảng 10^18 phép tính. Một máy tính hiện đại cần khoảng 1s để xử lý 10^9 câu lệnh => cần hơn 30 năm để sắp xếp.   + Với các thuật toán hiệu quả hơn (Merge Sort, Quick Sort) thì chỉ cần thực hiện khoảng 10^10 phép tính => khoảng 10s để sắp xếp.   + Máy tính hoạt động cần phần cứng và điện để hoạt động nên thời gian chạy càng lâu thì càng tốn kém, nhất là đối với những hệ thống lớn => thuật toán sinh ra để giải quyết các bài toán theo cách hiệu quả nhất * Độ phức tạp của thuật toán là thước đo độ hiệu quả của thuật toán dựa theo kích cỡ tập dữ liệu đầu vào.   + Có thể chia ra thành độ phức tạp về thời gian (thời gian thi hành của thuật toán) và độ phức tạp về không gian (kích cỡ bộ nhớ mà thuật toán cần để xử lý)   + Người ta thường dùng ký pháp O-lớn để nói về độ phức tạp của thuật toán. Ký pháp này thể hiện độ phức tạp trong trường hợp xấu nhất. Ký pháp này không thể hiện thời gian chạy thực tế của thuật toán, mà thể hiện mức độ thay đổi về thời gian chạy (hoặc bộ nhớ) của thuật toán khi kích cỡ dữ liệu đầu vào thay đổi. Tuy nhiên, ta vẫn có thể dùng ký pháp này để ước lượng thời gian chạy thực tế.   + Ví dụ: Một thuật toán sắp xếp có độ phức tạp về thời gian là O(N^2) thực thi trong 0.1s khi kích cỡ đầu vào là 10^4 số. Như vậy, khi kích cỡ đầu vào tăng gấp đôi (2\*10^4), ta có thể ước tính thời gian thực thi sẽ tăng gấp 2^2=4, tức khoảng 0.4s. * Phân tích độ phức tạp của thuật toán: Trong notebook bài học. | 45 | * [Reference ý nghĩa thuật toán](https://vnoi.info/wiki/translate/topcoder/The-Importance-of-Algorithm.md) * [Reference độ phức tạp thuật toán](https://vnoi.info/wiki/translate/topcoder/Computational-Complexity-Section-1.md) |
| **Giới thiệu cấu trúc dữ liệu**   * Cấu trúc dữ liệu là gì * Ý nghĩa | * Nhắc lại khái niệm: Cấu trúc dữ liệu là một cách tổ chức cao hơn của những kiểu dữ liệu cơ bản như int, float, boolean,... * Mỗi cấu trúc dữ liệu có một cách tổ chức khác nhau, tùy theo mục đích sử dụng. * Ý nghĩa: Ví dụ trong thư viện, ta cần tìm kiếm một quyển sách với chủ đề Lập trình Python căn bản.   + Các giá sách được xếp theo lĩnh vực. Trong lĩnh vực lập trình, sách lại được xếp theo ngôn ngữ lập trình, nhờ vậy mà việc lấy sách trở nên khá dễ dàng.   + Thử tưởng tượng thay vì tổ chức thành từng giá sách cụ thể, sách được chất thành từng đống ở khắp thư viện. Để tìm được đúng quyển sách sẽ phải mất hàng giờ, thậm chí rất nhiều ngày.   + => Cấu trúc dữ liệu hỗ trợ thuật toán thực thi khi xử lý. Đối với nhiều bài toán, cấu trúc dữ liệu đầu vào ảnh hưởng lớn đến độ phức tạp của thuật toán. | 10 | * [Reference ý nghĩa cấu trúc dữ liệu](https://vnoi.info/wiki/algo/data-structures/data-structures-overview.md) |
| **Mối quan hệ giữa thuật toán và cấu trúc dữ liệu** | * Cấu trúc dữ liệu có thể hỗ trợ thuật toán trong quá trình xử lý:   + Lưu trữ dữ liệu   + Lưu trữ các bước thực hiện thuật toán   + => CTDL có thể làm tăng hoặc giảm hiệu suất của thuật toán   + => Cần hiểu để chọn và sử dụng CTDL phù hợp cho bài toán | 10 |  |
| **Thực hành**   * Giải các bài cơ bản về thuật toán | * Một số bài toán:   + Dãy Fibonacci   + Kiểm tra số nguyên tố   + Tổng các số nguyên tố * Giảng viên hướng dẫn làm các bài toán trên, phân tích và so sánh độ phức tạp của từng phương pháp giải. Đối với bài tổng các số nguyên tố, phương pháp 2 là một dạng trade-off, đổi kích thước vùng nhớ lấy tốc độ xử lý. * Giảng viên giao bài tập về nhà | 40 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 2 - Thực Hành Online

*Created: April 05, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 2

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 3 - Các Cấu Trúc Dữ Liệu trong Python

*Created: April 08, 2021*

*Updated: June 02, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Cách hoạt động của các CTDL sẵn có trong Python
* Ứng dụng của từng CTDL
* Phân nhóm hai loại CTDL

1. **Kỹ năng**

* Phân tích và ứng dụng các CTDL sẵn có

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 15 |  |
| **Giới thiệu chung**   * Khái niệm CTDL * Các nhóm CTDL trong Python | * Nhắc lại VD trước về thư viện sách. * Khái niệm: CTDL là một cách sắp xếp và lưu trữ dữ liệu để dễ dàng xử lý và truy vấn. Một CTDL định nghĩa mối quan hệ giữa các thành phần dữ liệu trong nó và các phương thức để xử lý trên dữ liệu. * Nội dung bài học này gồm những CTDL sẵn có và thường dùng trong Python, các CTDL nâng cao sẽ được giới thiệu vào nửa sau khóa học. * Dẫn dắt vào các nhóm CTDL:   + Có thứ tự (ordered)   + Không có thứ tự (unordered) | 10 |  |
| **Nhóm CTDL có thứ tự**   * List * Tuple * String | * Đặc trưng, cách sử dụng của từng CTDL * Cách khởi tạo, truy vấn và một số phương thức phổ biến | 25 | * [Reference CTDL Python](https://realpython.com/python-data-structures/) * [Reference 2](https://docs.python.org/3/library/stdtypes.html) * [Các method của CTDL](https://www.w3schools.com/python/) |
| **Nhóm CTDL không có thứ tự**   * Set * Dictionary | * Đặc trưng, cách sử dụng của từng CTDL * Cách khởi tạo, truy vấn và một số phương thức phố biến | 15 |  |
| **So Sánh Hai Nhóm CTDL** | * So sánh đặc trưng của hai nhóm CTDL * So sánh tốc độ truy vấn | 15 |  |
| **Thực hành**   * Ứng dụng CTDL để giải một số bài toán | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng CTDL và làm bài tập trên CTDL đó. * Giảng viên giao bài tập về nhà | 40 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 3 - Thực Hành Online

*Created: April 08, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 3

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 4 - Thuật Toán Tìm Kiếm

*Created: April 12, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Các dạng bài toán tìm kiếm
* Các thuật toán tìm kiếm: Linear Search, Binary Search, Tìm kiếm Subset
* Phương pháp đệ quy

1. **Kỹ năng**

* Phân tích các bài toán tìm kiếm
* Ứng dụng và tùy chỉnh các thuật toán tìm kiếm đã học
* Ứng dụng phương pháp đệ quy

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 5 |  |
| **Giới thiệu về bài toán tìm kiếm** | * Tìm kiếm là một bài toán phổ biến trong lập trình:   + Nhập chữ để tìm bạn bè trên Facebook, tìm đoạn chat trên Facebook, tìm phần mềm trong thanh search của Windows taskbar, tìm kiếm trên Google, Youtube,...   + Cho danh sách khách hàng, tìm số điện thoại khách hàng dựa vào họ tên; cho doanh số bán hàng, tìm mặt hàng phổ biến nhất, các khách hàng trung thành,... * Bài học này bao gồm những bài toán tìm kiếm thông thường nhất và thuật toán để giải quyết những bài toán đó. * Tại sao phải học cách viết thuật toán khi Python và các thư viện như Numpy đã hỗ trợ?   + Mục đích của việc học thuật toán không phải để hoàn thành các việc như tìm kiếm, sắp xếp,... => những việc đó đã có thư viện hỗ trợ, và hỗ trợ rất tốt   + Học thuật toán để phát triển tư duy lập trình và tư duy thuật toán: cách phân tích một vấn đề từ dạng văn bản hay lời nói thành input, output và các bước để xử lý input thành output   + Học thuật toán để chọn cách lưu trữ và xử lý dữ liệu phù hợp để giải quyết các vấn đề lớn hơn (VD như Google không chỉ dùng một thuật toán tìm kiếm đơn giản mà là tổ hợp của nhiều thuật toán khác nhau), cũng như chọn thuật toán phù hợp để sử dụng. * Khi xử lý một bài toán tìm kiếm, ta quan tâm đến:   + Đặc trưng của thành phần cần tìm kiếm   + Đặc trưng của dữ liệu chứa thành phần đó | 15 |  |
| **Tìm kiếm tuần tự (Linear search)** | * Bài toán: Tìm kiếm vị trí của một phần tử dựa vào giá trị của nó trên một danh sách cho trước. * Ứng dụng: Tìm mã số khách hàng dựa vào họ tên; Cộng điểm thành viên dựa vào họ tên hoặc số điện thoại khách hàng. * Giảng viên mở link visualization để demo và giải thích * Nội dung: Trong notebook bài học. | 10 | * [Các thuật toán & Độ phức tạp](https://codelearn.io/sharing/5-thuat-toan-tim-kiem-moi-ltv-nen-biet) * [Code](https://www.geeksforgeeks.org/python-program-for-binary-search/) * [Visualization](https://www.cs.usfca.edu/~galles/visualization/Search.html) |
| **Tìm kiếm nhị phân (Binary search)** | * Bài toán: Tìm kiếm vị trí của một phần tử dựa vào giá trị của nó trên một danh sách đã được sắp xếp. * Giảng viên mở link visualization để demo và giải thích * Nội dung: Trong notebook bài học. | 30 |  |
| **Đệ quy**   * Khái niệm, ý nghĩa * Tìm kiếm nhị phân bằng đệ quy | * Khái niệm: Đệ quy là một thuật ngữ chỉ việc một hàm tự gọi chính nó. Thuật toán đệ quy thường dùng để giải quyết những bài toán có thể xử lý bằng cách đưa về các bài toán nhỏ hơn cùng loại. * Ví dụ với bài toán tính giai thừa * Tại sao phải sử dụng đệ quy? Ưu điểm và nhược điểm * Tìm kiếm nhị phân bằng đệ quy * Nội dung: Trong notebook bài học. | 30 | * [Đệ quy](https://techmaster.vn/posts/35006/khai-niem-de-quy-voi-java) |
| **Thực hành**   * Ứng dụng các thuật toán tìm kiếm | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng thuật toán và làm bài tập trên thuật toán đó. * Trong bài *2. Điểm Thi*, giảng viên giới thiệu cách dùng hàm trong Python để làm parameter cho function. * Giảng viên giao bài tập về nhà. | 30 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 4 - Thực Hành Online

*Created: April 12, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 4

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. * Nội dung một số bài: * Giải thích *Bài 1. Đếm Từ* là một dạng của Linear Search * Ở *Bài 3. Fibonacci*, giảng viên thử cho n=10^4 hoặc 10^5 để demo RecursionError. * Giải thích *Bài 4. Kiểm Tra Dãy Con* là một dạng bài đếm phân phối. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 5 - Thuật Toán Sắp Xếp (1)

*Created: April 15, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Bài toán sắp xếp
* Lớp thuật toán O(n^2): Selection Sort, Bubble Sort, Insertion Sort

1. **Kỹ năng**

* Phân tích các bài toán sắp xếp
* Ứng dụng và tùy chỉnh các thuật toán sắp xếp đã học

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 10 |  |
| **Giới thiệu về bài toán sắp xếp** | * Bài toán sắp xếp là bài toán sắp xếp một danh sách dữ liệu để dễ dàng truy vấn theo thứ tự tăng dần hoặc giảm dần. Tùy đặc trưng bài toán mà ta định nghĩa thế nào là tăng dần hoặc giảm dần. * Sắp xếp là một bài toán phổ biến trong lập trình:   + Sắp xếp danh sách lớp theo tên   + Sắp xếp các file trong thư mục theo tên, ngày tạo, ngày sửa, kích thước,...   + Sắp xếp thứ tự xuất hiện của các cửa hàng trên Grab, Foody * Bài học này giới thiệu các thuật toán cơ bản nhất để sắp xếp một danh sách. * Khi xử lý một bài toán sắp xếp, ta quan tâm đến các đặc trưng của dữ liệu cần sắp xếp:   + Phân bố của tập dữ liệu (ngẫu nhiên, gần như theo thứ tự, ngược thứ tự,...)   + Kiểu dữ liệu của các phần tử   + Cách tổ chức của tập dữ liệu và của các phần tử | 15 |  |
| **Selection Sort** | * Bài toán: Cho một dãy gồm n số nguyên, sắp xếp dãy trên theo thứ tự tăng dần. * Cách hoạt động * Visualization: giảng viên mở link để demo và giải thích * Độ phức tạp * Nội dung: Trong notebook bài học. | 20 | * [Selection Sort](https://www.geeksforgeeks.org/selection-sort/) * [Visualization](https://visualgo.net/en/sorting) |
| **Bubble Sort** | * Cách hoạt động * Visualization * Độ phức tạp * Nội dung: Trong notebook bài học. | 20 | * [Bubble & Insertion Sort](https://vnoi.info/wiki/algo/basic/sorting.md) |
| **Insertion Sort** | * Cách hoạt động * Visualization * Độ phức tạp * Nội dung: Trong notebook bài học. * ***Chú ý:*** Giảng viên tập trung giải thích về cách hoạt động của từng thuật toán. Tùy theo tốc độ học của lớp mà có thể giải thích kỹ về code hoặc nói qua và cho học viên tự nghiên cứu code mẫu. | 20 |  |
| **Thực hành**   * Ứng dụng các thuật toán sắp xếp | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng thuật toán và làm bài tập trên thuật toán đó. * Giảng viên giao bài tập về nhà. | 35 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 5 - Thực Hành Online

*Created: April 15, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 5

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 6 - Thuật Toán Sắp Xếp (2)

*Created: April 19, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Lớp thuật toán O(n.logn): Merge Sort, Quicksort

1. **Kỹ năng**

* Phân tích các bài toán sắp xếp
* Ứng dụng và tùy chỉnh các thuật toán sắp xếp đã học

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 10 |  |
| **Merge Sort** | * Cách hoạt động * Visualization: Giảng viên mở link để demo và giải thích * Độ phức tạp * Nội dung: Trong notebook bài học. | 20 | * [Merge Sort](https://www.geeksforgeeks.org/merge-sort/) * [Visualization](https://visualgo.net/en/sorting) |
| **Quicksort** | * Cách hoạt động * Visualization: Giảng viên mở link để demo và giải thích * Độ phức tạp * Nội dung: Trong notebook bài học. * ***Chú ý:*** Giảng viên tập trung giải thích về cách hoạt động của từng thuật toán. Tùy theo tốc độ học của lớp mà có thể giải thích kỹ về code hoặc nói qua và cho học viên tự nghiên cứu code mẫu. | 20 | * [Quicksort](https://www.geeksforgeeks.org/quick-sort/) |
| **Thực hành**   * Ứng dụng các thuật toán sắp xếp | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng thuật toán và làm bài tập trên thuật toán đó. * Giảng viên giao bài tập về nhà. | 35 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 6 - Thực Hành Online

*Created: April 19, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 6

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 7 - Thuật Toán Sắp Xếp (3)

*Created: April 22, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Lớp thuật toán O(n+k): Counting Sort
* So sánh các thuật toán sắp xếp

1. **Kỹ năng**

* Phân tích các bài toán sắp xếp
* Ứng dụng và tùy chỉnh các thuật toán sắp xếp đã học

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên review lại các nội dung chính của bài học trước. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 15 |  |
| **Counting Sort** | * Bài toán: Counting Sort giải quyết bài toán sắp xếp dãy có ít giá trị riêng biệt. * Cách hoạt động * Visualization: Giảng viên mở link để demo và giải thích * Độ phức tạp * Nội dung: Trong notebook bài học. | 35 | * [Counting Sort (1)](https://www.geeksforgeeks.org/counting-sort/) * [Counting Sort (2)](https://www.interviewcake.com/concept/java/counting-sort) * [Visualization](https://visualgo.net/en/sorting) |
| **So sánh các thuật toán sắp xếp** | * So sánh đặc trưng về độ phức tạp và cách hoạt động của các thuật toán * So sánh tương quan giữa độ phức tạp và thời gian chạy thực tế * Giảng viên có thể demo so sánh thời gian chạy giữa các thuật toán * Xem vui: Giảng viên mở video [visualization các thuật toán sắp xếp](https://www.youtube.com/watch?v=kPRA0W1kECg), có thể giới thiệu Bogosort | 30 |  |
| **Thực hành**   * Ứng dụng các thuật toán sắp xếp | * Giảng viên hướng dẫn làm các bài tập. * Giảng viên giao bài tập về nhà. | 40 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 7 - Thực Hành Online

*Created: April 22, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 7

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 8 - Kiểm Tra Giữa Khóa

*Created: April 26, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Ôn tập các kiến thức đã học: CTDL, Thuật Toán Tìm Kiếm, Thuật Toán Sắp Xếp

1. **Kỹ năng**

* Đánh giá bài toán và Phân tích thuật toán
* Ứng dụng và tùy chỉnh các thuật toán đã học

# Hậu Cần

* Sắp xếp chỗ ngồi sao cho mỗi học viên có không gian riêng để làm bài, tránh xem bài nhau

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Chuẩn bị** | * Hướng dẫn làm bài, đánh giá điểm và nộp bài. * Giảng viên mở link cho học viên download đề bài và nộp bài. Chú ý tạo link nộp bài sao cho học viên không thể xem bài của nhau. Có thể sử dụng Google Form, gửi mail hoặc cho mỗi học viên một link riêng để nộp. * Thời gian làm bài từ 1h45p đến 2h (tùy tình hình buổi học) | 5 | * Giảng viên chuẩn bị trước để không làm mất thời gian làm bài. * Các notebook đề bài, solution, test cases và evaluation nằm trong thư mục *Tests/Midterm Test* |
| **Làm bài** | * Học viên làm và nộp bài. | 115 |
| **Chấm bài** | * Giảng viên hoặc trợ giảng lưu các bài nộp vào cùng thư mục, sau đó làm theo hướng dẫn trong notebook *midterm-evaluation* để chấm từng bài. * Chú ý kiểm tra bài nộp có dùng thuật toán không. Trường hợp bài dùng random hoặc chỉ return cùng một giá trị (VD return True) thì toàn bộ test cases trên bài xem như fail. |  |

Lab 8 - Sửa Bài Kiểm Tra Giữa Khóa

*Created: April 26, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Học viên tìm hiểu cách giải các bài kiểm tra giữa khóa

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Giảng viên giao đáp án** | * Giảng viên giao đáp án trong notebook *midterm-solutions* và test case trong *midterm-testcases* để học viên tự nghiên cứu. | 120 | * Các notebook đề bài, solution, test cases và evaluation nằm trong thư mục *Tests/Midterm Test* |

Lesson 9 - CTDL: Stack và Queue

*Created: May 03, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* CTDL: Stack và Queue
* Ứng dụng của từng CTDL

1. **Kỹ năng**

* Phân tích các bài toán có thể sử dụng Stack và Queue
* Ứng dụng Stack và Queue

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên thông báo kết quả qua bảng điểm và giải đáp các thắc mắc về bài kiểm tra giữa khóa. * Đúc kết từ bài kiểm tra:   + Ôn tập kiến thức: CTDL, Thuật Toán Tìm Kiếm, Thuật Toán Sắp Xếp   + Luyện tập kỹ năng đánh giá bài toán và phân tích thuật toán   + Ứng dụng và tùy chỉnh các thuật toán đã học * Giảng viên nhắc lại các CTDL đã học, bao gồm *list, string, tuple, set, dict*. | 10 |  |
| **Stack**   * Cách hoạt động * Ứng dụng * Implement | * Cấu trúc, cách hoạt động * Ứng dụng của stack: * Giảng viên demo click qua nhiều link trên browser và back lại * Liên hệ giữa stack và đệ quy, stackoverflow, khử đệ quy * Code: sử dụng class *deque* * Nội dung trong notebook bài học. | 25 | * [Stack](https://www.geeksforgeeks.org/stack-in-python/) |
| **Queue**   * Cách hoạt động * Ứng dụng * Implement | * Cấu trúc, cách hoạt động * Ứng dụng của queue trong thực tế: đặt vé xem phim, sắp xếp người chơi vào phòng game online,... * Code: sử dụng class *deque*. * Nội dung trong notebook bài học. | 25 | * [Queue](https://www.geeksforgeeks.org/queue-in-python/) |
| **Deque**   * Cách hoạt động * Code | * Cách hoạt động * Code * Nội dung trong notebook bài học. | 10 |  |
| **Thực hành**   * Ứng dụng stack và queue | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng CTDL và làm bài tập trên CTDL đó. * Trong *Bài 2. Xử Lý Order*, giảng viên giải thích ý nghĩa chung của bài toán, cần chú ý phần nào trong code và phần nào bỏ qua, tránh đào sâu vào threading. * Giảng viên giao bài tập về nhà. Có thể hướng dẫn sơ qua về cách xử lý bài toán tính giá trị biểu thức. | 50 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 9 - Thực Hành Online

*Created: May 03, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 9

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 10 - CTDL: Tree và Graph

*Created: May 06, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* CTDL: Tree và Graph
* Ứng dụng của từng CTDL

1. **Kỹ năng**

* Phân tích các bài toán có thể sử dụng Tree và Graph
* Ứng dụng Tree và Graph

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên nhắc lại stack, queue; cách hoạt động và ứng dụng. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 10 |  |
| **Tree**   * Cấu trúc * Tính chất * Ứng dụng * Implement | * Cấu trúc, tính chất * Một số khái niệm: gốc, lá, cha-con, node * Ứng dụng * Code: tạo class *TreeNode*, đệ quy để duyệt qua các phần tử * Nội dung trong notebook bài học. | 30 | * [Binary Tree và cây nói chung](https://www.geeksforgeeks.org/binary-tree-data-structure/) * [Tree and Algorithms](https://runestone.academy/runestone/books/published/pythonds/Trees/toctree.html) |
| **Graph**   * Cấu trúc * Tính chất * Ứng dụng * Implement | * Cấu trúc, tính chất * Một số khái niệm: đỉnh, cạnh, đỉnh liền kề * Ứng dụng * Code:   + Sử dụng *dict* để lưu trữ dạng danh sách kề. Giảng viên có thể giới thiệu các cách lưu trữ khác như ma trận kề, danh sách cạnh nếu có thời gian.   + Kiểm tra 2 đỉnh có kết nối với nhau không bằng phương pháp duyệt đệ quy. * Mở rộng: đồ thị có trọng số, đồ thị có hướng, liên hệ giữa cây và đồ thị * Nội dung trong notebook bài học. | 40 | * [Graph](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/) |
| **Thực hành**   * Ứng dụng tree và graph | * Giảng viên hướng dẫn làm các bài tập. Có thể xen kẽ giữa dạy lý thuyết từng CTDL và làm bài tập trên CTDL đó. * Giảng viên hướng dẫn làm bài tập 2 và demo về cách chạy của thuật toán để chứng minh khoảng cách tìm được không đảm bảo là khoảng cách ngắn nhất. * Giảng viên giao bài tập về nhà. Có thể hướng dẫn sơ qua cây tìm kiếm nhị phân. | 40 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 10 - Thực Hành Online

*Created: May 06, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 10

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 90 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 11 - Thuật Toán Tìm Đường (1)

*Created: May 10, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Bài toán tìm đường
* Thuật toán DFS

1. **Kỹ năng**

* Phân tích các bài toán tìm đường
* Ứng dụng và tùy chỉnh DFS cho từng bài toán

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên nhắc lại tree, graph; cách hoạt động và ứng dụng. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 10 |  |
| **Giới thiệu về bài toán tìm đường** | * Tìm đường là bài toán tìm kiếm đường đi để duyệt qua tất cả các đỉnh trong một đồ thị. * Ứng dụng phổ biến nhất của thuật toán tìm đường là tìm đường đi trên bản đồ, như tìm đường cho các phương tiện khác nhau trên Google Maps * Tuy nhiên, ứng dụng của thuật toán tìm đường rất đa dạng:   + Tìm các bước để giải một khối rubik   + Tìm đường để thoát khỏi mê cung   + Xác định hướng truy đuổi cho các con “ma” trong game Pacman   + Xác định các vật thể khác nhau trong một bức ảnh   + Xác định màu để tô cho các nước trên bản đồ sao cho sử dụng ít màu nhất   + Xác định lộ trình khám phá cho robot trên hành tinh khác sao cho không bị mắc kẹt hoặc rơi xuống vực | 20 |  |
| **Depth-First Search (DFS)** | * Cách hoạt động, thứ tự duyệt => vì sao có tên là tìm kiếm theo chiều sâu * Visualization: giảng viên mở link để demo và giải thích trên đồ thị vô hướng * Độ phức tạp * Đặc điểm, ứng dụng * Nội dung: Trong notebook bài học. | 40 | * [DFS](https://cowboycoder.tech/article/ly-thuyet-do-thi-co-ban-phan-3-tim-kiem-theo-chieu-sau-tren-do-thi-depth-first-search-dfs) * [Visualization](https://visualgo.net/en/dfsbfs) |
| **Thực hành**   * Ứng dụng DFS | * Giảng viên hướng dẫn làm các bài tập. * Giảng viên giao bài tập về nhà, review và giải thích sơ qua bài tập. | 50 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 11 - Thực Hành Online

*Created: May 10, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 11

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 12 - Thuật Toán Tìm Đường (2)

*Created: May 13, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Bài toán tìm đường
* Thuật toán BFS

1. **Kỹ năng**

* Phân tích các bài toán tìm đường
* Ứng dụng và tùy chỉnh BFS cho từng bài toán

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Nhắc lại bài cũ** | * Giảng viên nhắc lại thuật toán DFS; cách hoạt động, đặc điểm và ứng dụng. * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. | 10 |  |
| **Breadth-First Search (BFS)** | * Cách hoạt động, thứ tự duyệt => vì sao có tên là tìm kiếm theo chiều rộng * Visualization: giảng viên mở link để demo và giải thích trên đồ thị vô hướng * Độ phức tạp * Đặc điểm, ứng dụng * Nội dung: Trong notebook bài học. | 45 | * [BFS](https://cowboycoder.tech/article/ly-thuyet-do-thi-co-ban-tim-kiem-theo-chieu-rong-tren-do-thi-breadth-first-search-bfs) * [Visualization](https://visualgo.net/en/dfsbfs) |
| **Thực hành**   * Ứng dụng BFS | * Giảng viên hướng dẫn làm các bài tập. * Giảng viên giao bài tập về nhà, review và giải thích sơ qua bài tập. | 65 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 12 - Thực Hành Online

*Created: May 13, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 12

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 120 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên sửa bài** | * Giảng viên hướng dẫn solution của từng bài, phân tích thuật toán để pass tất cả test case. | Buổi sau |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 13 - Ôn Tập Cuối Khóa

*Created: May 17, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Ôn tập các kiến thức đã học về Thuật toán và CTDL

1. **Kỹ năng**

* Ứng dụng tổng hợp thuật toán, CTDL và các kĩ thuật đã học

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Ôn tập tổng hợp** | * Giảng viên sửa các bài tập chưa giải quyết xong và các bài lab. * Giảng viên ôn tập tất cả nội dung đã học:   + Các CTDL trong Python   + CTDL: Stack, Queue, Tree, Binary Tree, Graph   + Thuật Toán Tìm Kiếm   + Thuật Toán Sắp Xếp   + Thuật Toán Tìm Đường * CTDL: cấu trúc, cách truy vấn và ứng dụng của từng CTDL * Thuật toán: ý tưởng, cách hoạt động, độ phức tạp và ứng dụng của từng thuật toán * Khuyến khích học viên đặt câu hỏi và giải đáp thắc mắc * Nội dung trong notebook bài học | 60 |  |
| **Chuẩn bị thi đấu cuối khóa** | * Giảng viên giới thiệu nội quy và hình thức thi:   + Thi đấu theo đội từ 1 đến 3 thành viên   + Thời gian thi: 2 giờ.   + Học viên nhận link thi và notebook đề thi tiếng Việt.   + Các nội quy khác và cách tính điểm nằm trong đề thi mẫu. * Giảng viên hướng dẫn thi thử:   + Học viên tạo account trên hackerrank.   + Học viên download notebook đề tiếng Việt, vào link đề thi mẫu.   + Học viên đọc đề bài, làm và nộp bài. * Đề thi mẫu được dùng chung cho tất cả các lớp, không cần tạo contest mới trên hackerrank. Tuy nhiên, nếu cần tạo mới đề thi mẫu, giảng viên có thể xem file doc hướng dẫn setup. | 50 | * [Đề thi mẫu](https://www.hackerrank.com/mindx-sample-contest-1621345079) * Notebook đề thi tiếng Việt, cách setup contest và video hướng dẫn tổ chức thi trong thư mục *Tests/ Final Test* * GIảng viên xem trước nội dung thi cuối khóa và làm quen với nền tảng thi đấu *hackerrank* |
| **Thực hành**   * Ứng dụng tổng hợp kiến thức đã học | * Giảng viên giao bài tập về nhà. | 10 | * Bài tập về nhà trong thư mục *Homeworks* |

Lab 13 - Thực Hành Online

*Created: May 17, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Ôn tập các kiến thức trong Lesson 13

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Học viên thực hành** | * Học viên làm các bài tập thực hành và tự kiểm tra trên test cases cho sẵn. | 110 | * Các bài tập và đáp án trong thư mục *Labs* |
| **Giảng viên giao đáp án** | * Giảng viên giao đáp án bài 13 để học viên tự tìm hiểu và ôn tập cho bài kiểm tra cuối khóa | 10 |  |

# RULE

*Giảng viên trực lớp follow đủ các công việc được giao và hỗ trợ học viên về mặt kỹ thuật, hạn chế support về kiến thức hoặc gợi ý, đưa ra câu hỏi mở để tăng sự chủ động và tự học của học viên*.

Lesson 14 - Kiểm Tra Cuối Khóa

*Created: May 24, 2021*

*Updated: June 03, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

1. **Kiến thức**

* Ôn tập tất cả kiến thức đã học

1. **Kỹ năng**

* Đánh giá bài toán và Phân tích thuật toán
* Ứng dụng và tùy chỉnh các thuật toán đã học
* Tạo sân chơi, cho học viên tiếp cận với lập trình thi đấu

# Hậu Cần

* Sắp xếp chỗ ngồi theo nhóm, đảm bảo mỗi nhóm tách biệt nhau
* Có thể xếp bàn ghế đối diện hoặc xoay các hướng khác nhau để tăng độ kịch tính
* Đảm bảo mỗi nhóm có thể theo dõi màn hình chiếu trong quá trình thi đấu

# Nội Dung

| **Nội dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Chuẩn bị** | * Giảng viên tạo contest trên hackerrank theo hướng dẫn trong file *Hướng Dẫn Setup Contests.docx*. Contest được tạo nên giống hệt với [contest mẫu](https://www.hackerrank.com/mindx-csi-final-contest-1621345079). * Giảng viên xem video hướng dẫn tổ chức cuộc thi: *Final Test - Hướng dẫn tổ chức.wmv* |  | * Giảng viên chuẩn bị trước để không làm mất thời gian làm bài. |
| **Làm bài** | * Giảng viên mở link contest, cho học viên download notebook đề bài tiếng Việt. * Các nhóm làm và nộp từng bài. * Giảng viên mở leaderboard trên màn hình chiếu trong suốt thời gian làm bài. | 120 | * Các notebook đề bài và đáp án nằm trong thư mục *Tests/ Final Test* |
| **Đánh giá điểm, xếp hạng** | * Giảng viên mở bảng xếp hạng cuộc thi và công bố kết quả. |  | * Bảng xếp hạng được công bố ngay sau cuộc thi |

Lab 14 - Sửa Bài Kiểm Tra Cuối Khóa

*Created: May 24, 2021*

*Updated: June 18, 2021*

*Contributors: Nguyễn Phú Hưng*

# Mục Tiêu

* Học viên tìm hiểu cách giải các bài kiểm tra cuối khóa

# Nội Dung

| **Nội Dung** | **Mô tả hoạt động** | **Thời gian** | **Ghi chú** |
| --- | --- | --- | --- |
| **Giảng viên giao đáp án** | * Giảng viên giao đáp án trong notebook *Đáp Án* để học viên tự nghiên cứu. | 120 | * Các notebook đề bài và đáp án nằm trong thư mục *Tests/ Final Test* |