



Đề cương môn học

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**  
**(Data Structures and Algorithms)**

Số tín chỉ	4 (3.2.7)			MSMH	CO2003
Số tiết	Tổng: <b>75</b>	LT: <b>45</b>	TH:	TN: <b>30</b>	BTL/TL: <b>x</b>
Môn ĐA, TT, LV					
Tỉ lệ đánh giá	BT: <b>15%</b>	TN: <b>10%</b>	KT: <b>0%</b>	BTL/TL: <b>25%</b>	Thi: <b>50%</b>
Hình thức đánh giá	<ul style="list-style-type: none"><li>- <b>Bài tập:</b> sinh viên làm trước bài tập ở nhà; bài tập được chấm theo cách chấm được nêu trong cột cuối cùng của bảng danh mục các bài tập (thực hành), được trình bày ở phần sau, gần cuối bản đề cương.</li><li>- <b>Thí nghiệm:</b> sinh viên làm trước các bài thí nghiệm ở nhà; các bài thí nghiệm được chấm theo cách chấm được nêu trong cột cuối cùng của bảng danh mục các thí nghiệm, được trình bày ở phần sau, gần cuối bản đề cương.</li><li>- <b>Thi:</b> viết và trắc nghiệm, 120 phút</li></ul>				
Môn tiên quyết	Không				
Môn học trước	Kỹ thuật Lập trình				CO1011
Môn song hành	Không				
CTĐT ngành	Khoa Học Máy Tính và Kỹ Thuật Máy Tính				
Trình độ đào tạo	Đại học				
Cấp độ môn học	Cấp độ 2 (dạy cho sinh viên năm 2)				
Ghi chú khác					

**1. Mục tiêu của môn học**

Môn học nhằm cung cấp cho sinh viên khả năng sử dụng các cấu trúc dữ liệu nền tảng. Môn học cũng hướng dẫn sinh viên hiểu, phân tích và đánh giá được các giải thuật làm việc với các cấu trúc dữ liệu đó.

**Aims:**

This course is to provide students abilities to use fundamental data structures. It also helps students understanding, analyzing, and evaluating algorithms associated with those data structures.

**2. Nội dung tóm tắt môn học**

- Ôn lại về lập trình, các kiểu dữ liệu trong C/C++, đặc biệt là cấu trúc và con trỏ.
- Giới thiệu về độ phức tạp tính toán và đệ qui.
- Các cấu trúc dữ liệu và sự phân tích chúng: danh sách; chồng và hàng; cây, cây nhị phân, cây nhị phân tìm kiếm, AVL và đa phân; heap; giải thuật sắp xếp; bảng băm; và đồ thị.

### Course outline:

- Review on programming and data types in C/C++, especially, struct and pointer.
- Basics of computational complexity and recursive algorithms.
- Data structures and their analysis: list, stack and queue, tree, binary tree, binary search tree, AVL and M-ways tree, sorting, hashing table, and graph

### 3. Tài liệu học tập

Sách, Giáo trình chính:

- [1]. “*Data Structures: a Pseudocode Approach with C++*”, R.F.Gilberg and B.A. Forouzan, Thomson Learning Inc., 2001.

Sách tham khảo:

- [1] “*Data Structures and Algorithms in C++*”, A. Drozdek, Thomson Learning Inc., 2005.  
[2] “*C/C++: How to Program*”, 7<sup>th</sup> Ed. – Paul Deitel and Harvey Deitel, Prentice Hall, 2012.  
[3] Internet.

### 4. Hiểu biết, kỹ năng, thái độ cần đạt được sau khi học môn học

STT	Chuẩn đầu ra môn học	CDIO
<b>L.O.1</b>	<i>Phân tích giải thuật</i> <b>L.O.1.1</b> – Định nghĩa được các khái niệm độ phức tạp và độ phức tạp trong các trường hợp “tốt nhất”, “xấu nhất”, và “trung bình”. <b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp. <b>L.O.1.3</b> – Liệt kê được, cho được ví dụ và so sánh được các lớp độ phức tạp, như, hằng số, log, tuyến tính, etc. <b>L.O.1.4</b> – Nhận thức được sự cân bằng giữa bộ nhớ và thời gian trong giải thuật. <b>L.O.1.5</b> – Mô tả được các chiến lược thiết kế giải thuật và giải quyết bài toán.	
<b>L.O.2</b>	<i>Sử dụng cấu trúc dữ liệu danh sách, chồng và hàng</i> <b>L.O.2.1</b> – Phát họa được bằng hình ảnh cho: (a) danh sách hiện thực bằng mảng và bằng liên kết (con trỏ); (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic). <b>L.O.2.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho: (a) danh sách hiện thực bằng mảng và bằng liên kết; (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic). <b>L.O.2.3</b> – Liệt kê được các phương thức cần thiết cho từng cấu trúc như danh sách, chồng và hàng đợi; cũng như mô tả được chúng bằng mã giả (mức logic). <b>L.O.2.4</b> – Hiện thực được các cấu trúc danh sách, chồng và hàng đợi bằng C/C++ (mức physics) <b>L.O.2.5</b> – Sử dụng được danh sách, chồng, và hàng để giải quyết bài toán thực, cũng như cân nhắc chọn lựa kiểu hiện thực tối ưu. <b>L.O.2.6</b> – Phân tích được và làm thí nghiệm đánh giá các phương thức đã hỗ trợ cho các cấu trúc danh sách, chồng, và hàng.	

<b>L.O.3</b>	<i>Sử dụng cấu trúc cây</i>	
	<p><b>L.O.3.1</b> – Phát họa được bằng hình ảnh cho các cây tiêu biểu, như, cây nhị phân, cây nhị phân đầy đủ, cây nhị phân cân bằng, cây AVL, cây đa phân, v.v. (mức logic).</p> <p><b>L.O.3.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho các loại cây (mức logic)</p> <p><b>L.O.3.3</b> – Liệt kê được các phương thức cần thiết cho cho các cấu trúc cây; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.3.4</b> – Chỉ ra được và cho ví dụ minh họa về tầm quan trọng của tính cân bằng trong cây.</p> <p><b>L.O.3.5</b> – Chỉ ra được và vẽ hình minh họa về tất cả các trường mất cân bằng trong cây AVL và cây B, cũng như thực hiện từng bước để tái cân bằng chúng trên hình vẽ (mức logic).</p> <p><b>L.O.3.6</b> – Hiện thực được các cấu trúc cây nhị phân và cây AVL bằng C/C++</p> <p><b>L.O.3.7</b> – Sử dụng được cây nhị phân và cây AVL để giải quyết bài toán thực, đặc biệt là liên quan đến tìm kiếm.</p> <p><b>L.O.3.8</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho các cấu trúc cây nhị phân và cây AVL.</p>	
<b>L.O.4</b>	<i>Sử dụng Heap</i>	
	<p><b>L.O.4.1</b> – Chỉ ra được những ứng dụng cần đến Heap</p> <p><b>L.O.4.2</b> – Phác họa được bằng hình ảnh cho cấu trúc Heap và nêu ra sự liên quan đến lưu trữ ở dạng mảng.</p> <p><b>L.O.4.3</b> – Liệt kê được các phương thức cần thiết cho cho cấu trúc heap; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.4.4</b> – Phác họa được bằng hình ảnh các phương thức để đảm bảo tính chất của cấu trúc Heap khi đưa vào hay lấy ra phần tử trong heap (mức logic).</p> <p><b>L.O.4.5</b> – Hiện thực được cấu trúc heap bằng C/C++.</p> <p><b>L.O.4.6</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho cấu trúc Heap.</p>	
<b>L.O.5</b>	<i>Sử dụng bảng băm</i>	
	<p><b>L.O.5.1</b> – Vẽ được hình minh họa một bảng băm cùng với khái niệm về khóa, đựng độ và giải quyết đựng độ.</p> <p><b>L.O.5.2</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các hàm băm cơ bản.</p> <p><b>L.O.5.3</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các phương thức giải quyết đựng độ.</p> <p><b>L.O.5.4</b> – Hiện thực được cấu trúc bảng băm bằng C/C++.</p> <p><b>L.O.5.5</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho cấu trúc bảng băm.</p>	
<b>L.O.6</b>	<i>Phát triển các giải thuật sắp xếp</i>	
	<p><b>L.O.6.1</b> – Minh họa được bằng hình vẽ từng bước hoạt động của các giải thuật sắp xếp.</p> <p><b>L.O.6.2</b> – Mô tả được bằng mã giả cho các giải thuật sắp xếp.</p> <p><b>L.O.6.3</b> – Hiện thực được các giải thuật sắp xếp bằng C/C++ .</p> <p><b>L.O.6.4</b> – Phân tích được và làm thực nghiệm đánh giá các giải thuật sắp</p>	

	xếp. <b>L.O.6.5</b> – Sử dụng được giải thuật sắp xếp trong bài toán thực.	
<b>L.O.7</b>	<i>Hiểu biết cơ bản về đồ thị</i>	
	<p><b>L.O.7.1</b> – Phát họa được bằng hình ảnh cho các khái niệm như đồ thị liên thông và không liên thông, đồ thị có hướng và không hướng, chu trình, v.v.</p> <p><b>L.O.7.2</b> – Vẽ được hình minh họa và mô tả cấu trúc lưu trữ cho đồ thị ở các dạng ma trận kề và danh sách kề bằng mã giả (mức logic).</p> <p><b>L.O.7.3</b> – Liệt kê được các phương thức cần thiết cho các cấu trúc đồ thị; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.7.4</b> – Minh họa được bằng hình ảnh các phương pháp duyệt đồ thị cơ bản (depth first and bread-first).</p> <p><b>L.O.7.5</b> – Hiện thực được cấu trúc lưu trữ đồ thị bằng C/C++.</p> <p><b>L.O.7.6</b> – Hiện thực được các phương pháp duyệt nói trên bằng C/C++ và sử dụng chúng giải quyết bài toán thực.</p> <p><b>L.O.7.7</b> – Minh họa bằng hình vẽ từng bước hoạt động của các giải thuật tìm đường ngắn nhất bằng Dijkstra và giải thuật tìm cây phủ tối thiểu bằng giải thuật Prim.</p>	
<b>L.O.8</b>	<i>Sử dụng đệ quy</i>	
	<p><b>L.O.8.1</b> – Mô tả được các thành phần cơ bản của một giải thuật đệ quy.</p> <p><b>L.O.8.2</b> – Vẽ được cây mô tả các lần gọi hàm và giá trị của các tham số được truyền vào các hàm đó.</p> <p><b>L.O.8.3</b> – Cho được ví dụ về một hàm gọi đệ quy viết bằng C/C++.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.8.5</b> – Làm được thí nghiệm để so sánh cách tiếp cận đệ quy và cách lặp.</p> <p><b>L.O.8.6</b> – Cho được ví dụ minh họa sự liên quan giữa Backtracking và đệ quy.</p>	

Index	Course learning outcomes	CDIO
<b>L.O.1</b>	<i>Analyze algorithms</i>	
	<p><b>L.O.1.1</b> – Define concept “computational complexity” and its special cases, best, average, and worst.</p> <p><b>L.O.1.2</b> – Analyze algorithms and use Big-O notation to characterize the computational complexity of algorithms composed by using the following control structures: sequence, branching, and iteration (not recursion).</p> <p><b>L.O.1.3</b> – List, give examples, and compare complexity classes, for examples, constant, linear, etc.</p> <p><b>L.O.1.4</b> – Be aware of the trade-off between space and time in solutions</p> <p><b>L.O.1.5</b> – Describe strategies in algorithm design and problem solving</p>	
<b>L.O.2</b>	<i>Use data structures: list, stack and queue</i>	
	<p><b>L.O.2.1</b> – Depict the following concepts: (a) array list and linked list, including single link and double links, and multiple links; (b) stack; and (c) queue and circular queue.</p> <p><b>L.O.2.2</b> – Describe storage structures by using pseudocode for: (a) array list and linked list, including single link and double links, and multiple</p>	

	<p>links; (b) stack; and (c) queue and circular queue.</p> <p><b>L.O.2.3</b> – List necessary methods supplied for list, stack, and queue, and describe them using pseudocode.</p> <p><b>L.O.2.4</b> – Implement list, stack, and queue using C/C++.</p> <p><b>L.O.2.5</b> – Use list, stack, and queue for problems in real-life, and choose an appropriate implementation type (array vs. link).</p> <p><b>L.O.2.6</b> – Analyze the complexity and develop experiment (program) to evaluate the efficiency of methods supplied for list, stack, and queue.</p>	
<b>L.O.3</b>	<p><i>Use tree structure</i></p> <p><b>L.O.3.1</b> – Depict the following concepts: binary tree, complete binary tree, balanced binary tree, AVL tree, multi-way tree, etc.</p> <p><b>L.O.3.2</b> – Describe the storage structure for tree structures using pseudocode.</p> <p><b>L.O.3.3</b> – List necessary methods supplied for tree structures, and describe them using pseudocode.</p> <p><b>L.O.3.4</b> – Identify the importance of “balanced” feature in tree structures and give examples to demonstrate it.</p> <p><b>L.O.3.5</b> – Identify cases in which AVL tree and B-tree are unbalanced, and demonstrate methods to resolve all the cases step-by-step using figures.</p> <p><b>L.O.3.6</b> – Implement binary tree and AVL tree using C/C++</p> <p><b>L.O.3.7</b> – Use binary tree and AVL tree to solve problems in real-life, especially related to searching techniques.</p> <p><b>L.O.3.8</b> – Analyze the complexity and develop experiment (program) to evaluate methods supplied for tree structures.</p>	
<b>L.O.4</b>	<p><i>Use heap structure</i></p> <p><b>L.O.4.1</b> – List some applications of Heap.</p> <p><b>L.O.4.2</b> – Depict heap structure and relate it to array.</p> <p><b>L.O.4.3</b> – List necessary methods supplied for heap structure, and describe them using pseudocode.</p> <p><b>L.O.4.4</b> – Depict the working steps of methods that maintain the characteristics of heap structure for the cases of adding/removing elements to/from heap.</p> <p><b>L.O.4.5</b> – Implement heap using C/C++.</p> <p><b>L.O.4.6</b> – Analyze the complexity and develop experiment (program) to evaluate methods supplied for heap structures.</p>	
<b>L.O.5</b>	<p><i>Use hash structure</i></p> <p><b>L.O.5.1</b> – Depict the following concepts: hashing table, key, collision, and collision resolution.</p> <p><b>L.O.5.2</b> – Describe hashing functions using pseudocode and give examples to show their algorithms.</p> <p><b>L.O.5.3</b> – Describe collision resolution methods using pseudocode and give examples to show their algorithms.</p> <p><b>L.O.5.4</b> – Implement hashing tables using C/C++.</p> <p><b>L.O.5.5</b> – Analyze the complexity and develop experiment (program) to evaluate methods supplied for hashing tables.</p>	
<b>L.O.6</b>	<p><i>Use sorting</i></p> <p><b>L.O.6.1</b> – Depict the working steps of sorting algorithms step-by-steps.</p>	

	<b>L.O.6.2</b> – Describe sorting algorithms by using pseudocode. <b>L.O.6.3</b> – Implement sorting algorithms using C/C++ . <b>L.O.6.4</b> – Analyze the complexity and develop experiment (program) to evaluate sorting algorithms. <b>L.O.6.5</b> – Use sorting algorithms for problems in real-life.	
<b>L.O.7</b>	<i>Basically understand graph structure</i>	
	<b>L.O.7.1</b> – Depict the following concepts: connected graph, disconnected graph, direct/undirect graph, etc. <b>L.O.7.2</b> – Depict storage structures for graph and describe graph using pseudocode in the cases of using adjacency matrix and adjacency list. <b>L.O.7.3</b> – List necessary methods supplied for graph, and describe them using pseudocode. <b>L.O.7.4</b> – Depict basic traversal methods step-by-step (depth first and breadth-first). <b>L.O.7.5</b> – Implement storage structures for graphs using C/C++. <b>L.O.7.6</b> – Implement basic traversal methods using C/C++. <b>L.O.7.7</b> – Depict the working steps of Dijkstra and Prim step-by-step.	
<b>L.O.8</b>	<i>Use recursion</i>	
	<b>L.O.8.1</b> – Describe the basic components of recursive algorithms (functions). <b>L.O.8.2</b> – Draw trees to illustrate callings and the value of parameters passed to them for recursive algorithms. <b>L.O.8.3</b> – Give examples for recursive functions written in C/C++. <b>L.O.8.4</b> – Develop recursive implementations for methods supplied for the following structures: list, tree, heap, searching, and graphs. <b>L.O.8.5</b> – Develop experiment (program) to compare the recursive and the iterative approach. <b>L.O.8.6</b> – Give examples to relate recursion to backtracking technique.	

## 5. Hướng dẫn cách học - chi tiết cách đánh giá môn học

Hướng dẫn cách học:

- Tài liệu học tập bao gồm: đề cương môn học, slide bài giảng, bài tập, bài thí nghiệm, và bài tập lớn được lưu trữ trên máy chủ quản lý tư liệu học tập của khoa (trường) . Sinh viên tải về, in ra và mang theo khi lên lớp học.
- Sinh viên cần làm thêm các bài tập và các bài thực hành. Sinh viên nên tham gia làm bài tập online trên hệ thống máy chủ nói trên, cũng như sử dụng hệ thống này để trao đổi với sinh viên khác, TA, và giảng viên.
- Sinh viên nên đi học đầy đủ và làm bài tập trong quá trình học sẽ giúp tiết kiệm thời gian trong quá trình ôn thi giữa kỳ và cuối kỳ.
- Đối với phần thực hành và bài tập, sinh viên tham gia đầy đủ các buổi thí nghiệm và nộp lại báo cáo thí nghiệm ngay cuối giờ thí nghiệm.

Chi tiết cách đánh giá môn học:

- Thực hành (15%):
- Bài tập (10%):
- Bài tập lớn (25%):

- Thi cuối kỳ (50%)

## 6. Dự kiến danh sách Cán bộ tham gia giảng dạy

- TS. Lê Thành Sách
- TS. Huỳnh Tường Nguyên
- Th.S. Trần Giang Sơn
- Th.S. Nguyễn Trung Trực
- Th.S. Võ Thanh Hùng
- Th.S. Vương Bá Thịnh

## 7. Nội dung chi tiết

### Nội dung phân lý thuyết

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
<i>Xem bảng phân tiết</i>	<b>Chương 1. Giới thiệu</b> 1.1. Giới thiệu về môn học 1.2. Các khái niệm: dữ liệu, kiểu dữ liệu, kiểu dữ liệu trừu tượng, cấu trúc dữ liệu, giải thuật. 1.3. Ôn tập về struct, class, con trỏ, và mảng. <b>1.4. Bài tập</b> <i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i>		- Giảng lý thuyết - Bài tập trên lớp	- Bài tập - Bài thí nghiệm - Bài tập lớn - Bài thi
	<b>Chương 2. Độ phức tạp giải thuật</b> 2.1. Khái niệm độ phức tạp 2.2. Ký hiệu Big-O và các trường hợp 2.3. Các bài toán và các độ phức tạp thường gặp 2.4. Giới thiệu về P và NP <b>2.5. Bài tập</b> <i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i>	<b>L.O.1.1</b> – Định nghĩa được các khái niệm độ phức tạp và độ phức tạp trong các trường hợp hợp “tốt nhất”, “xấu nhất”, và “trung bình”. <b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp. <b>L.O.1.3</b> – Liệt kê được, cho được ví dụ và so sánh được các lớp độ phức tạp, như, hằng số, log, tuyến tính, etc. <b>L.O.1.4</b> – Nhận thức được sự cân bằng giữa bộ nhớ và thời gian trong giải thuật. <b>L.O.1.5</b> – Mô tả được các chiến lược thiết kế giải thuật và giải quyết bài toán.	- Giảng lý thuyết - Bài tập trên lớp	- Bài tập - Bài thí nghiệm - Bài tập lớn - Bài thi
	<b>Chương 3. Độ quy</b>	<b>L.O.8.1</b> – Mô tả được các thành phần cơ bản	- Giảng lý thuyết - Bài tập trên lớp	- Bài tập - Bài thí nghiệm

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
	<p>3.1 Đệ quy và các thành phần cơ bản của giải thuật đệ quy.</p> <p>3.2 Đệ quy và C/C++.</p> <p>3.3 Tính chất của đệ quy.</p> <p>3.4 Thiết kế giải thuật đệ qui</p> <p>3.5 Đệ quy và kỹ thuật quay lui</p> <p><b>3.6 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p>của một giải thuật đệ quy.</p> <p><b>L.O.8.2</b> – Vẽ được cây mô tả các lần gọi hàm và giá trị của các tham số được truyền vào các hàm đó.</p> <p><b>L.O.8.3</b> – Cho được ví dụ về một hàm gọi đệ quy viết bằng C/C++.</p> <p><b>L.O.8.5</b> – Làm được thí nghiệm để so sánh cách tiếp cận đệ quy và cách lặp.</p> <p><b>L.O.8.6</b> – Cho được ví dụ minh họa sự liên quan giữa Backtracking và đệ quy.</p>		<ul style="list-style-type: none"> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>
	<p><b>Chương 4. Danh sách</b></p> <p>4.1 Khái niệm và ứng dụng</p> <p>4.2 Hiện thực bằng mảng (array)</p> <p>4.3 Hiện thực bằng liên kết đơn</p> <p>4.4 Các dạng liên kết phức tạp khác</p> <p>4.5 Đánh giá các dạng hiện thực</p> <p><b>4.1. Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8</i></p>	<p><b>L.O.2.1</b> – Phát họa được bằng hình ảnh cho: (a) danh sách hiện thực bằng mảng và bằng liên kết (con trỏ); (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic).</p> <p><b>L.O.2.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho: (a) danh sách hiện thực bằng mảng và bằng liên kết; (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic).</p> <p><b>L.O.2.3</b> – Liệt kê được các phương thức cần thiết cho từng cấu trúc như danh sách, chồng và hàng đợi; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.2.4</b> – Hiện thực được các cấu trúc danh sách, chồng và hàng đợi bằng C/C++ (mức physics)</p> <p><b>L.O.2.5</b> – Sử dụng được danh sách, chồng, và hàng để giải quyết bài toán thực, cũng như cân nhắc chọn lựa kiểu hiện thực tối ưu.</p> <p><b>L.O.2.6</b> – Phân tích được và làm thí nghiệm đánh giá các phương</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>



Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p>thức đã hỗ trợ cho các cấu trúc danh sách, chồng, và hàng.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<p><b>Chương 5. Chồng và Hàng</b></p> <p>5.1 Các ứng dụng của chồng &amp; hàng</p> <p>5.2 Các tác vụ cơ bản trên chồng</p> <p>5.3 Hiện thực chồng bằng danh sách liên kết và mảng</p> <p>5.4 Các tác vụ cơ bản trên hàng</p> <p>5.5 Hiện thực hàng bằng danh sách liên kết và mảng</p> <p><b>5.6 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p><b>L.O.2.1</b> – Phát họa được bằng hình ảnh cho: (a) danh sách hiện thực bằng mảng và bằng liên kết (con trỏ); (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic).</p> <p><b>L.O.2.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho: (a) danh sách hiện thực bằng mảng và bằng liên kết; (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic).</p> <p><b>L.O.2.3</b> – Liệt kê được các phương thức cần thiết cho từng cấu trúc như danh sách, chồng và hàng đợi; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.2.4</b> – Hiện thực được các cấu trúc danh sách, chồng và hàng đợi bằng C/C++ (mức physics)</p> <p><b>L.O.2.5</b> – Sử dụng được danh sách, chồng, và hàng để giải quyết bài toán thực, cũng như cân nhắc chọn lựa kiểu hiện thực tối ưu.</p> <p><b>L.O.2.6</b> – Phân tích được và làm thí nghiệm đánh giá các phương</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p>thức đã hỗ trợ cho các cấu trúc danh sách, chồng, và hàng.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<b>Kiểm tra giữa kỳ</b>			
	<p><b>Chương 6. Cây</b></p> <p>6.1 Các khái niệm căn bản về cây và ứng dụng</p> <p>6.2 Cây nhị phân</p> <p>6.3 Cấu trúc lưu trữ và các phương thức</p> <p>6.4 Cây nhị phân tìm kiếm</p> <p>6.5 Cây biểu thức</p> <p><b>6.6 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p><b>L.O.3.1</b> – Phát họa được bằng hình ảnh cho các cây tiêu biểu, như, cây nhị phân, cây nhị phân đầy đủ, cây nhị phân cân bằng, cây AVL, cây đa phân, v.v. (mức logic).</p> <p><b>L.O.3.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho các loại cây (mức logic)</p> <p><b>L.O.3.3</b> – Liệt kê được các phương thức cần thiết cho cho các cấu trúc cây; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.3.4</b> – Chỉ ra được và cho ví dụ minh họa về tầm quan trọng của tính cân bằng trong cây.</p> <p><b>L.O.3.5</b> – Chỉ ra được và vẽ hình minh họa về tất cả các trường mất cân bằng trong cây AVL và cây B, cũng như thực hiện từng bước để tái cân bằng chúng trên hình vẽ (mức logic).</p> <p><b>L.O.3.6</b> – Hiện thực được các cấu trúc cây nhị phân và cây AVL bằng C/C++</p> <p><b>L.O.3.7</b> – Sử dụng được cây nhị phân và</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p>cây AVL để giải quyết bài toán thực, đặc biệt là liên quan đến tìm kiếm.</p> <p><b>L.O.3.8</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho các cấu trúc cây nhị phân và cây AVL.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<p><b>Chương 7. Cây AVL</b></p> <p>7.1 Cây AVL: Khái niệm, ưu điểm và ứng dụng</p> <p>7.2 Cấu trúc lưu trữ và các phương thức</p> <p>7.3 Cân bằng cây AVL</p> <p>7.4 Cây đa phân: khái niệm, ưu điểm và ứng dụng</p> <p>7.5 Cây B-Tree: khái niệm</p> <p>7.6 Cân bằng cây B-Tree</p> <p><b>7.7 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 16 giờ</i></p>	<p><b>L.O.3.1</b> – Phát họa được bằng hình ảnh cho các cây tiêu biểu, như, cây nhị phân, cây nhị phân đầy đủ, cây nhị phân cân bằng, cây AVL, cây đa phân, v.v. (mức logic).</p> <p><b>L.O.3.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho các loại cây (mức logic)</p> <p><b>L.O.3.3</b> – Liệt kê được các phương thức cần thiết cho cho các cấu trúc cây; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.3.4</b> – Chỉ ra được và cho ví dụ minh họa về tầm quan trọng của tính cân bằng trong cây.</p> <p><b>L.O.3.5</b> – Chỉ ra được và vẽ hình minh họa về tất cả các trường mất cân bằng trong cây AVL và cây B, cũng như thực hiện từng bước để tái cân bằng chúng trên hình vẽ (mức logic).</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p><b>L.O.3.6</b> – Hiện thực được các cấu trúc cây nhị phân và cây AVL bằng C/C++</p> <p><b>L.O.3.7</b> – Sử dụng được cây nhị phân và cây AVL để giải quyết bài toán thực, đặc biệt là liên quan đến tìm kiếm.</p> <p><b>L.O.3.8</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho các cấu trúc cây nhị phân và cây AVL.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<p><b>Chương 8. Heap</b></p> <p>8.1 Khái niệm và ứng dụng</p> <p>8.2 Cấu trúc lưu trữ và các phép toán</p> <p><b>8.3 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 4 giờ</i></p>	<p><b>L.O.4.1</b> – Chỉ ra được những ứng dụng cần đến Heap</p> <p><b>L.O.4.2</b> – Phác họa được bằng hình ảnh cho cấu trúc Heap và nêu ra sự liên quan đến lưu trữ ở dạng mảng.</p> <p><b>L.O.4.3</b> – Liệt kê được các phương thức cần thiết cho cấu trúc heap; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.4.4</b> – Phác họa được bằng hình ảnh các phương thức để đảm bảo tính chất của cấu trúc Heap khi đưa vào hay lấy ra phần tử trong heap (mức logic).</p> <p><b>L.O.4.5</b> – Hiện thực được cấu trúc heap bằng C/C++.</p> <p><b>L.O.4.6</b> – Phân tích được và làm thực nghiệm đánh giá được</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p>các phương thức đã hỗ trợ cho cấu trúc Heap.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<p><b>Chương 9. Bảng băm</b></p> <p>9.1 Khái niệm và ứng dụng</p> <p>9.2 Các hàm băm</p> <p>9.3 Các phương pháp giải quyết đụng độ</p> <p><b>9.4 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p><b>L.O.5.1</b> – Vẽ được hình minh họa một bảng băm cùng với khái niệm về khóa, đụng độ và giải quyết đụng độ.</p> <p><b>L.O.5.2</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các hàm băm cơ bản.</p> <p><b>L.O.5.3</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các phương thức giải quyết đụng độ.</p> <p><b>L.O.5.4</b> – Hiện thực được cấu trúc bảng băm bằng C/C++.</p> <p><b>L.O.5.5</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho cấu trúc bảng băm.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>
	<p><b>Chương 10. Sắp xếp</b></p> <p>10.1 Khái niệm và ứng dụng</p> <p>10.2 Các giải thuật chèn</p> <p>10.3 Các giải thuật chọn</p> <p>10.4 Các giải thuật trao đổi</p> <p>10.5 Các giải thuật sắp xếp ngoài</p> <p>10.6 Bài tập</p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p><b>L.O.6.1</b> – Minh họa được bằng hình vẽ từng bước hoạt động của các giải thuật sắp xếp.</p> <p><b>L.O.6.2</b> – Mô tả được bằng mã giả cho các giải thuật sắp xếp.</p> <p><b>L.O.6.3</b> – Hiện thực được các giải thuật sắp xếp bằng C/C++ .</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p><b>L.O.6.4</b> – Phân tích được và làm thực nghiệm đánh giá các giải thuật sắp xếp.</p> <p><b>L.O.6.5</b> – Sử dụng được giải thuật sắp xếp trong bài toán thực.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<p><b>Chương 11. Đồ thị</b></p> <p>11.1 Khái niệm và ứng dụng</p> <p>11.2 Cấu trúc lưu trữ và các phương thức</p> <p>11.3 Cây phủ tối thiểu</p> <p>11.4 Đường đi ngắn nhất</p> <p>11.5 Các bài toán ứng dụng trên đồ thị</p> <p><b>11.6 Bài tập</b></p> <p><i>Yêu cầu tự học đ/v sinh viên: 8 giờ</i></p>	<p><b>L.O.7.1</b> – Phát họa được bằng hình ảnh cho các khái niệm như đồ thị liên thông và không liên thông, đồ thị có hướng và không hướng, chu trình, v.v.</p> <p><b>L.O.7.2</b> – Vẽ được hình minh họa và mô tả cấu trúc lưu trữ cho đồ thị ở các dạng ma trận kề và danh sách kề bằng mã giả (mức logic).</p> <p><b>L.O.7.3</b> – Liệt kê được các phương thức cần thiết cho các cấu trúc đồ thị; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.7.4</b> – Minh họa được bằng hình ảnh các phương pháp duyệt đồ thị cơ bản (depth first and bread-first).</p> <p><b>L.O.7.5</b> – Hiện thực được cấu trúc lưu trữ đồ thị bằng C/C++.</p> <p><b>L.O.7.6</b> – Hiện thực được các phương pháp duyệt nói trên bằng C/C++ và sử dụng chúng giải quyết bài toán thực.</p> <p><b>L.O.7.7</b> – Minh họa</p>	<ul style="list-style-type: none"> <li>- Giảng lý thuyết</li> <li>- Bài tập trên lớp</li> </ul>	<ul style="list-style-type: none"> <li>- Bài tập</li> <li>- Bài thí nghiệm</li> <li>- Bài tập lớn</li> <li>- Bài thi</li> </ul>

Tuần /Tiết	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy và học	Hoạt động đánh giá
		<p>bằng hình vẽ từng bước hoạt động của các giải thuật tìm đường ngắn nhất bằng Dijkstra và giải thuật tìm cây phủ tối thiểu bằng giải thuật Prim.</p> <p><b>L.O.8.4</b> – Phát triển được giải thuật đệ quy cho các phương thức cần thiết của các cấu trúc: danh sách, cây, heap, tìm kiếm trên cây và tìm kiếm nhị phân, và đồ thị.</p> <p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p>		
	<b>Review</b>			
**	<p><b>Nội dung giới hạn cho kiểm tra giữa kỳ (tập trung)</b></p> <p>Chương 1 – 5</p> <p>Ước tính số giờ SV cần chuẩn bị để kiểm tra giữa kỳ: 12</p>			
**	<p><b>Nội dung thi cuối kỳ (tập trung)</b></p> <p>Các chương còn lại.</p> <p>Ước tính số giờ SV cần chuẩn bị để thi cuối kỳ: 16</p>			

**Bảng phân tiết**

Chương	1	2	3	4	5	6	7	8	9	10	11
Số tiết	3	3	3	6	2	4	6	2	4	6	6

**Nội dung phần thực hành (bài tập)**

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
1	<p><b>Bài tập số 1: Phân tích độ phức tạp</b></p> <p>1.1. Phân tích độ phức tạp của giải thuật lặp và sử dụng ký hiệu Big-O để mô tả.</p> <p>1.2. Các lớp độ phức tạp và so sánh</p> <p>1.3. Phát triển giải thuật cho một số bài toán tiêu biểu và phân tích chúng</p> <p>Yêu cầu tự học đ/v sinh viên:</p>	<p><b>L.O.1.2</b> – Phân tích được các giải thuật và sử dụng được ký hiệu “Big O” để ghi ra độ phức tạp của giải thuật cấu thành từ các cấu trúc điều khiển: tuần tự, rẽ nhánh và lặp.</p> <p><b>L.O.1.3</b> – Liệt kê được, cho được ví dụ và so sánh được các lớp độ phức tạp, như, hằng số, log, tuyến tính, etc.</p>	<p>➤ Thầy/Cô:</p> <ul style="list-style-type: none"> <li>Hướng dẫn làm bài tiêu biểu.</li> <li>Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)</li> </ul> <p>➤ Sinh viên:</p> <ul style="list-style-type: none"> <li>Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ</li> </ul>	Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần, và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
		<b>L.O.1.5</b> – Mô tả được các chiến lược thiết kế giải thuật và giải quyết bài toán.	quản lý tư liệu học tập). - Lên bảng làm các bài tập theo yêu cầu của giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi.	
2	<b>Bài tập số 2: Sử dụng danh sách</b> 2.1. Mô tả ý tưởng hiện thực danh sách bằng hình ảnh và bằng mã giả. 2.2. Viết mã giả và thực hiện từng bước cho các phương thức trên danh sách, chồng, và hàng Yêu cầu tự học đ/v sinh viên:	<b>L.O.2.1</b> – Phát họa được bằng hình ảnh cho: (a) danh sách hiện thực bằng mảng và bằng liên kết (con trỏ); (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic). <b>L.O.2.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho: (a) danh sách hiện thực bằng mảng và bằng liên kết; (b) cho chồng; và (c) cho hàng đợi và hàng đợi vòng (mức logic). <b>L.O.2.3</b> – Liệt kê được các phương thức cần thiết cho từng cấu trúc như danh sách, chồng và hàng đợi; cũng như mô tả được chúng bằng mã giả (mức logic).	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Lên bảng làm các bài tập theo yêu cầu của giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi.	Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần, và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên
3	<b>Bài tập số 3: Sử dụng cây</b> 3.1. Xác định các tính chất của cây: cây đầy đủ, cân bằng, hệ số cân bằng, etc. 3.2. Thực hiện từng bước cân bằng cây AVL bằng hình ảnh Yêu cầu tự học đ/v sinh viên:	<b>L.O.3.1</b> – Phát họa được bằng hình ảnh cho các cây tiêu biểu, như, cây nhị phân, cây nhị phân đầy đủ, cây nhị phân cân bằng, cây AVL, cây đa phân, v.v. (mức logic). <b>L.O.3.2</b> – Viết được bằng mã giả mô tả cấu trúc lưu trữ cho các loại cây (mức logic) <b>L.O.3.5</b> – Chỉ ra được và vẽ hình minh họa về tất cả các trường mất cân bằng trong cây AVL và cây B, cũng như thực hiện từng bước để tái cân bằng chúng trên hình vẽ (mức logic).	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Lên bảng làm các bài tập theo yêu cầu của giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi.	Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần, và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên
4	<b>Bài tập số 4: Sử dụng Bảng băm và Heap</b> 4.1. Đảm bảo tính chất của heap bằng	<b>L.O.4.2</b> – Phác họa được bằng hình ảnh cho cấu trúc Heap và	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu.	Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần,



Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
	<p>hình ảnh</p> <p>4.2. Phân tích độ phức tạp của các phương thức</p> <p>4.3. Xây dựng bảng băm với các hàm băm tiêu biểu</p> <p>4.4. Theo vết quá trình giải quyết đưng độ của một giải thuật</p> <p>Yêu cầu tự học đ/v sinh viên:</p>	<p>nêu ra sự liên quan đến lưu trữ ở dạng mảng.</p> <p><b>L.O.4.3</b> – Liệt kê được các phương thức cần thiết cho cấu trúc heap; cũng như mô tả được chúng bằng mã giả (mức logic).</p> <p><b>L.O.4.4</b> – Phác họa được bằng hình ảnh các phương thức để đảm bảo tính chất của cấu trúc Heap khi đưa vào hay lấy ra phần tử trong heap (mức logic).</p> <p><b>L.O.5.2</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các hàm băm cơ bản.</p> <p><b>L.O.5.3</b> – Mô tả được bằng mã giả và cho ví dụ minh họa cho các phương thức giải quyết đưng độ.</p> <p><b>L.O.4.6</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho cấu trúc Heap.</p> <p><b>L.O.5.5</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho cấu trúc bảng băm.</p>	<p>- Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)</p> <p>➤ Sinh viên:</p> <p>- Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ quản lý tư liệu học tập).</p> <p>- Lên bảng làm các bài tập theo yêu cầu của giảng viên.</p> <p>- Nộp lời giải cho giảng viên ở cuối buổi.</p>	<p>và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên</p>
5	<p><b>Bài tập số 5: Sử dụng các giải thuật sắp xếp</b></p> <p>5.1. Thực hiện từng bước quá trình sắp xếp cho các giải thuật.</p> <p>5.2. Phân tích các giải thuật sắp xếp và đặc trưng bởi ký hiệu Big-O.</p> <p>5.3. Sử dụng mã giả mô tả giải thuật sắp xếp</p> <p>Yêu cầu tự học đ/v sinh viên:</p>	<p><b>L.O.6.1</b> – Minh họa được bằng hình vẽ từng bước hoạt động của các giải thuật sắp xếp.</p> <p><b>L.O.6.2</b> – Mô tả được bằng mã giả cho các giải thuật sắp xếp.</p> <p><b>L.O.6.4</b> – Phân tích được và làm thực nghiệm đánh giá các giải thuật sắp xếp.</p>	<p>➤ Thầy/Cô:</p> <p>- Hướng dẫn làm bài tiêu biểu.</p> <p>- Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)</p> <p>➤ Sinh viên:</p> <p>- Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ quản lý tư liệu học tập).</p> <p>- Lên bảng làm các bài tập theo yêu cầu của giảng viên.</p> <p>- Nộp lời giải cho giảng viên ở cuối buổi.</p>	<p>Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần, và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên</p>

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
6	<b>Bài tập số 6: Sử dụng đồ thị</b> 6.1. Mô tả cấu trúc lưu trữ cho đồ thị 6.2. Thực hiện theo vết từng bước quá trình duyệt đồ thị. 6.3. Minh họa bằng hình ảnh các giải thuật Dijkstra và Prim Yêu cầu tự học đ/v sinh viên:	<b>L.O.7.2</b> – Vẽ được hình minh họa và mô tả cấu trúc lưu trữ cho đồ thị ở các dạng ma trận kề và danh sách kề bằng mã giả (mức logic). <b>L.O.7.4</b> – Minh họa được bằng hình ảnh các phương pháp duyệt đồ thị cơ bản (depth first and bread-first). <b>L.O.7.7</b> – Minh họa bằng hình vẽ từng bước hoạt động của các giải thuật tìm đường ngắn nhất bằng Dijkstra và giải thuật tìm cây phủ tối thiểu bằng giải thuật Prim.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trên bảng hay làm trên giấy (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà vào giấy (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Lên bảng làm các bài tập theo yêu cầu của giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi.	Đánh giá trên thái độ tham gia các buổi bài tập, sự chuyên cần, và chất lượng bài giải của sinh viên trên bảng hay bài giải nộp cho giảng viên
	<b>Review</b>			
**	<b>Nội dung báo cáo tiểu luận/thực hành</b> Yêu cầu đ/v sinh viên: Sinh viên làm trước bài tập ở nhà và nộp lại bài tập cho giảng viên cuối buổi.			
**	<b>Nội dung giới hạn cho kiểm tra giữa kỳ (tập trung)</b> Không có			
**	<b>Nội dung thi cuối kỳ (tập trung)</b> (Ước tính số giờ SV cần để chuẩn bị cho kỳ thi: )			

### Nội dung phần thí nghiệm

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
1	<b>Bài thực hành số 1</b> 1.1. Luyện tập sử dụng các kiểu dữ liệu cơ bản và kiểu do người dùng định nghĩa trong C/C++ 1.2. Luyện tập sử dụng con trỏ và tham khảo trong C/C++ 1.3. Viết các chương trình đệ quy trong C/C++ Yêu cầu tự học đ/v sinh viên:	<b>L.O.8.3</b> – Give examples for recursive functions written in C/C++. <b>L.O.8.6</b> – Give examples to relate recursion to backtracking technique.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
2	<b>Bài thực hành số 2</b> 2.1. Hiện thực danh sách ở dạng array và liên kết. 2.2. Làm thí nghiệm đánh giá giữa tìm kiếm tuần tự và tìm kiếm nhị phân. 2.3. Làm thí nghiệm đánh giá giữa lập và đệ quy. Yêu cầu tự học đ/v sinh viên:	<b>L.O.8.5</b> – Làm được thí nghiệm để so sánh cách tiếp cận đệ quy và cách lập. <b>L.O.2.4</b> – Hiện thực được các cấu trúc danh sách, chồng và hàng đợi bằng C/C++ (mức physics)	thông máy chủ. ➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên) ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (đề lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.
3	<b>Bài thực hành số 3</b> 3.1. Hiện thực chồng và hàng 3.2. Sử dụng chồng và hàng để giải quyết bài toán thực tế Yêu cầu tự học đ/v sinh viên:	<b>L.O.2.4</b> – Hiện thực được các cấu trúc danh sách, chồng và hàng đợi bằng C/C++ (mức physics) <b>L.O.2.5</b> – Sử dụng được danh sách, chồng, và hàng để giải quyết bài toán thực, cũng như cân nhắc chọn lựa kiểu hiện thực tối ưu.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên) ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (đề lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.
4	<b>Bài thực hành số 4</b> 4.1. Hiện thực cây nhị phân tìm kiếm và cây AVL 4.2. Làm thí nghiệm đánh giá giữa tìm kiếm với cây nhị phân, AVL và tìm kiếm tuần tự. Yêu cầu tự học đ/v sinh viên:	<b>L.O.3.6</b> – Hiện thực được các cấu trúc cây nhị phân và cây AVL bằng C/C++ <b>L.O.3.8</b> – Phân tích được và làm thực nghiệm đánh giá được các phương thức đã hỗ trợ cho các cấu trúc cây nhị phân và cây AVL.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên) ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (đề lấy từ hệ thống máy chủ quản lý tư	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
			liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.	
5	<b>Bài thực hành số 5</b> 5.1. Hiện thực cấu trúc Heap. 5.2. Ứng dụng Heap để giải quyết bài toán thực Yêu cầu tự học đ/v sinh viên:	<b>L.O.4.5</b> – Hiện thực được cấu trúc heap bằng C/C++.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.
6	<b>Bài thực hành số 6</b> 6.1. Hiện thực bảng băm. 6.2. Ứng dụng bảng băm để giải quyết bài toán thực Yêu cầu tự học đ/v sinh viên:	<b>L.O.5.4</b> – Hiện thực được cấu trúc bảng băm bằng C/C++.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên)  ➤ Sinh viên: - Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (để lấy từ hệ thống máy chủ quản lý tư liệu học tập). - Trình bày lời giải cho giảng viên. - Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.
7	<b>Bài thực hành số 7</b> 7.1. Hiện thực các giải thuật sắp xếp 7.2. Làm thí nghiệm đánh giá các giải thuật sắp xếp Yêu cầu tự học đ/v sinh viên:	<b>L.O.6.3</b> – Hiện thực được các giải thuật sắp xếp bằng C/C++ . <b>L.O.6.4</b> – Phân tích được và làm thực nghiệm đánh giá các giải thuật sắp xếp.	➤ Thầy/Cô: - Hướng dẫn làm bài tiêu biểu. - Phân tích và chỉnh sửa bài giải của sinh viên làm trước ở nhà (lấy ngẫu nhiên)  ➤ Sinh viên:	Đánh giá trên thái độ tham gia thí nghiệm, chuyên cần, coding style, và bài giải của sinh viên trên file nộp ở máy chủ hay file nộp trực tiếp tại lớp.

Tuần	Nội dung	Chuẩn đầu ra chi tiết	Hoạt động dạy/học	Hoạt động đánh giá
			<ul style="list-style-type: none"> <li>- Làm trước bài tập ở nhà và chép file mang theo, cũng như nộp lên hệ thống máy chủ (đề lấy từ hệ thống máy chủ quản lý tư liệu học tập).</li> <li>- Trình bày lời giải cho giảng viên.</li> <li>- Nộp lời giải cho giảng viên ở cuối buổi hay nộp trên hệ thống máy chủ.</li> </ul>	
	<b>Review</b>			
**	<b>Nội dung báo cáo tiểu luận/thực hành</b> Yêu cầu đ/v sinh viên: Sinh viên làm trước ở nhà và nộp lên máy chủ, cũng như có thể chỉnh sửa và nộp trực tiếp cho giảng viên			
**	<b>Nội dung giới hạn cho kiểm tra giữa kỳ (tập trung)</b> Không có			
**	<b>Nội dung thi cuối kỳ (tập trung)</b> (Ước tính số giờ SV cần để chuẩn bị cho kỳ thi: )			

## 8. Thông tin liên hệ

Bộ môn/Khoa phụ trách	Bộ Môn Khoa Học Máy Tính – Khoa KH&KT Máy Tính
Văn phòng	
Điện thoại	38647256 - 5839
Giảng viên phụ trách	Lê Thành Sách
Email	LTSACH@hcmut.edu.vn

*Tp. Hồ Chí Minh, ngày tháng năm 2015*

**TRƯỞNG KHOA**

**CHỦ NHIỆM BỘ MÔN**

**CB PHỤ TRÁCH LẬP ĐỀ CƯƠNG**