

**ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**THUYẾT MINH
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP TRƯỜNG
NĂM 2022**

**TÊN ĐỀ TÀI
Xây dựng video bài giảng môn Cấu trúc dữ liệu và giải thuật
MÃ SỐ T2022-VD37**

Chủ nhiệm đề tài: ThS. Nguyễn Thị Hương

THÁI NGUYÊN, NĂM 2022

**THUYẾT MINH ĐỀ TÀI
KHOA HỌC VÀ CÔNG NGHỆ CẤP TRƯỜNG NĂM 2022**

1. TÊN ĐỀ TÀI: Xây dựng video bài giảng môn Cấu trúc dữ liệu và giải thuật		2. MÃ SỐ: T2022-VD37	
3. LĨNH VỰC NGHIÊN CỨU		4. LOẠI HÌNH NGHIÊN CỨU	
Khoa học Tự nhiên <input type="checkbox"/>	Khoa học Kỹ thuật và Công nghệ <input checked="" type="checkbox"/>	Cơ bản	Ứng dụng
Khoa học Y, dược <input type="checkbox"/>	Khoa học Nông nghiệp <input type="checkbox"/>	Triển khai	
Khoa học Xã hội <input type="checkbox"/>	Khoa học Nhân văn <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. THỜI GIAN THỰC HIỆN DỰ KIẾN: 12 tháng			
Từ tháng 4 năm 2022 đến tháng 4 năm 2023.			
6. CHỦ NHIỆM ĐỀ TÀI			
Họ và tên: Nguyễn Thị Hương		Học vị: Thạc sĩ	
Chức danh khoa học:		Năm sinh: 1974	
Địa chỉ cơ quan: Khoa Điện tử, ĐHKTCN		Điện thoại di động: 0988143690	
Điện thoại cơ quan:		Fax: 0208 3847403	
E-mail: huongktpm@tnut.edu.vn			
7. NHỮNG THÀNH VIÊN THAM GIA NGHIÊN CỨU ĐỀ TÀI			
TT	Họ và tên	Đơn vị công tác và lĩnh vực chuyên môn	Nội dung nghiên cứu cụ thể được giao
1			Chữ ký
8. ĐƠN VỊ PHỐI HỢP CHÍNH			
Tên đơn vị trong và ngoài nước	Nội dung phối hợp nghiên cứu		Họ và tên người đại diện đơn vị

9. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU THUỘC LĨNH VỰC CỦA ĐỀ TÀI Ở TRONG VÀ NGOÀI NƯỚC

9.1. Trong nước: Hiện tại trên youtube, mạng xã hội có rất nhiều video bài giảng liên quan đến học phần Cấu trúc dữ liệu và giải thuật. Tuy nhiên, các video này thường không đầy đủ theo đề cương giảng dạy cho sinh viên trong trường, có video quá dài, có video quá ngắn, không xây dựng được tư duy tổng thể cho sinh viên. Vì vậy, việc xây dựng các video trình bày về các nội dung chính của bài học theo đề cương chương trình giảng dạy là vô cùng cần thiết cho sinh viên đặc biệt là trong giai đoạn giảng dạy online.

9.2. Ngoài nước (*phân tích, đánh giá tình hình nghiên cứu thuộc lĩnh vực của đề tài trên thế giới, liệt kê danh mục các công trình nghiên cứu, tài liệu có liên quan đến đề tài được trích dẫn khi đánh giá tổng quan*)

9.3. Danh mục các công trình đã công bố thuộc lĩnh vực của đề tài của chủ nhiệm và những thành viên tham gia nghiên cứu (*họ và tên tác giả; bài báo; ấn phẩm; các yếu tố về xuất bản*)

a) Của chủ nhiệm đề tài

- Nguyễn Thị Hương; Bài giảng Cấu trúc dữ liệu và giải thuật, NXB Khoa học Kỹ Thuật Năm 2010

b) Của các thành viên tham gia nghiên cứu

(*Những công trình được công bố trong 5 năm gần nhất*)

10. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Trước sự ảnh hưởng của dịch Covid – 19, Nhà trường đã triển khai việc giảng dạy online cho sinh viên toàn trường. Để duy trì và đảm bảo chất lượng đào tạo thì công tác xây dựng tài liệu giảng dạy online, nhằm giúp sinh viên chuẩn bị bài trước giờ học là vô cùng quan trọng.

Học phần Cấu trúc dữ liệu và giải thuật là học phần cơ bản được giảng dạy cho sinh viên ngành. Đây là học phần sinh viên cần học tập, làm bài tập, xây dựng phần mềm trên máy tính vì vậy việc xây dựng video trực quan, sinh động, giúp sinh viên hiểu bài là rất cần thiết.

11. MỤC TIÊU ĐỀ TÀI

Đề tài xây dựng 24 video bài giảng tương ứng với 45 tiết học của sinh viên. Mỗi tiết học được cô đọng giảng dạy để sinh viên có thể nắm được tổng thể nội dung của bài học và chuẩn bị bài trước khi đến lớp.

12. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU

12.1. Đối tượng nghiên cứu: Xây dựng video bài giảng cho học phần Cấu trúc dữ liệu và giải thuật

12.2. Phạm vi nghiên cứu: Xây dựng 24 video tương ứng với 45 tiết giảng dạy

13. CÁCH TIẾP CẬN, PHƯƠNG PHÁP NGHIÊN CỨU

13.1. Cách tiếp cận: Video sẽ bám sát vào đề cương, bài giảng học phần

13.2. Phương pháp nghiên cứu: Quay video và giảng trên phần mềm Power Point

14. NỘI DUNG NGHIÊN CỨU VÀ TIẾN ĐỘ THỰC HIỆN

14.1. Nội dung nghiên cứu (*Mô tả chi tiết những nội dung nghiên cứu của đề tài*)

Chương 1: Mở đầu

Chương 2: Thiết kế và phân tích

Chương 3: Đề quy và giải thuật đề quy

Chương 4: Mạng và danh sách

Chương 5: Danh sách móc nối

Chương 6: Cây

Chương 7: Lý thuyết đồ thị

Chương 8: Sắp xếp

Chương 9: Tìm kiếm

14.2. Tiến độ thực hiện

ST T	Các nội dung, công việc thực hiện	Sản phẩm	Thời gian (bắt đầu-kết thúc)	Người thực hiện
1	Xây dựng thuyết minh đề tài	Bản thuyết minh và dự toán kinh phí của đề tài	04/2022	Nguyễn Thị Hương
2	Xây dựng các file bài giảng trình chiếu Powerpoint cho môn học	File trình chiếu Powerpoint	05/2022 – 06/2022	Nguyễn Thị Hương
3	Ghi hình các bài giảng chương 1,2	Video	07/2022 – 08/2022	Nguyễn Thị Hương
4	Ghi hình các bài giảng chương 3,4	Video	09/2022 – 10/2022	Nguyễn Thị Hương
5	Ghi hình các bài giảng chương 5,6	Video	11/2022	Nguyễn Thị Hương
6	Ghi hình các bài giảng chương 7	Video	12/2022	Nguyễn Thị Hương
7	Ghi hình các bài giảng chương 8,9	Video	01/2023	Nguyễn Thị Hương

8	Hậu kỳ chỉnh sửa các video	Các video hoàn thiện	02/2023	Nguyễn Thị Hương
9	Viết báo cáo tổng kết	Quyển báo cáo tổng kết đề tài	03/2023	Nguyễn Thị Hương

15. SẢN PHẨM

Stt	Tên sản phẩm	Số lượng	Yêu cầu chất lượng sản phẩm (mô tả chi tiết chất lượng sản phẩm đạt được như nội dung, hình thức, các chỉ tiêu, thông số kỹ thuật, ...)
I	Sản phẩm khoa học (Các công trình khoa học sẽ được công bố: sách, bài báo khoa học, ...)		
1.1			
II	Sản phẩm đào tạo (cử nhân, thạc sĩ, tiến sĩ, ...)		
2.1	Video bài giảng học phần cấu trúc dữ liệu và giải thuật	24 video	- Đủ số lượng video. - Bài giảng bám sát Đề cương chi tiết học phần Cấu trúc dữ liệu và giải thuật. - Chất lượng hình ảnh và âm thanh được đảm bảo.
III	Sản phẩm ứng dụng		
3.1			

16. PHƯƠNG THỨC CHUYỂN GIAO KẾT QUẢ NGHIÊN CỨU VÀ ĐỊA CHỈ ỨNG DỤNG

16.1. Phương thức chuyển giao: Bộ video bài giảng sẽ được chuyển giao cho Bộ môn Tin học Công Nghiệp, Khoa Điện tử, trường Đại học Kỹ thuật Công nghiệp.

16.2. Địa chỉ ứng dụng: Trường Đại học Kỹ thuật Công nghiệp.

17. TÁC ĐỘNG VÀ LỢI ÍCH MANG LẠI CỦA KẾT QUẢ NGHIÊN CỨU

17.1. Đối với lĩnh vực giáo dục và đào tạo:

- Có thể sử dụng như các bài giảng trong Đào tạo trực tuyến, từ xa
- Sinh viên có thể học trên các video theo thời gian chủ động và không giới hạn số lần. Do đó sinh viên có thể xem trước bài giảng và đến giờ học trao đổi, thảo luận sâu hơn với giáo viên.

17.2. Đối với lĩnh vực khoa học và công nghệ có liên quan

17.3. Đối với phát triển kinh tế-xã hội

17.4. Đối với tổ chức chủ trì và các cơ sở ứng dụng kết quả nghiên cứu

- Góp phần xây dựng, hoàn thiện bộ học liệu phục vụ giảng dạy / học tập trong Nhà trường.

- Đảm bảo và nâng cao chất lượng đào tạo trong Nhà trường trong bối cảnh dịch bệnh Covid-19.

- Có thể khai thác, sử dụng phục vụ cho công tác Đào tạo từ xa.

18. KINH PHÍ THỰC HIỆN ĐỀ TÀI

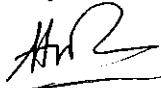
Tổng kinh phí: 5.400.000 VND

Bằng chữ: Năm triệu bốn trăm ngàn đồng chẵn./

(Dự toán chi tiết các mục chi đính kèm có xác nhận của các đơn vị liên quan.)

Ngày tháng năm 2022

Chủ nhiệm đề tài



Nguyễn Thị Hương

PHÒNG KHCN&HTQT

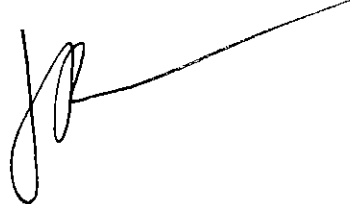


PGS.TS. Phạm Thành Long

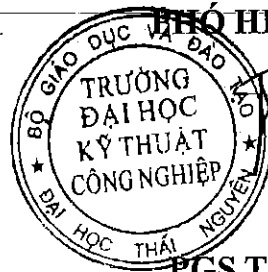
HỘI ĐỒNG KHOA KHOA ĐIỆN TỬ

KT. HIỆU TRƯỞNG

HIỆU TRƯỞNG



PGS.TS. Đào Huy Du



PGS.TS. Vũ Ngọc Pi

**ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

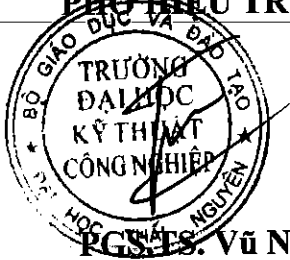
**BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP TRƯỜNG**

**XÂY DỰNG VIDEO BÀI GIẢNG
MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

Mã số: T2022-VD37

Xác nhận của tổ chức chủ trì

**KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG**



PGS.TS. Vũ Ngọc Pi

Chủ nhiệm đề tài

(ký, họ tên)

ThS. Nguyễn Thị Hương

Thái Nguyên, tháng 10 năm 2023

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: **Xây dựng video bài giảng môn Cấu trúc dữ liệu và giải thuật**
- Mã số: **T2022-VD37**
- Chủ nhiệm đề tài: **ThS. Nguyễn Thị Hương**
- Cơ quan chủ trì: **Trường Đại học Kỹ thuật Công nghiệp**
- Thời gian thực hiện: **Tháng 3/2022 – /2023**

2. Mục tiêu: Đề tài xây dựng 24 video bài giảng. Mỗi video được cô đọng giảng dạy từ 8 - 20 phút, tùy nội dung cần trình bày để sinh viên có thể nắm được tổng thể nội dung của bài học và chuẩn bị bài trước khi đến lớp.

3. Kết quả nghiên cứu: 24 video

4. Sản phẩm:

- Sản phẩm đào tạo: 24 video làm tài liệu học tập cho sinh viên
- Sản phẩm khoa học:
- Sản phẩm ứng dụng:

5. Hiệu quả: Nâng cao chất lượng giảng dạy và học tập học phần Cấu trúc dữ liệu và giải thuật

6. Khả năng áp dụng và phương thức chuyển giao kết quả nghiên cứu: dùng làm tài liệu học tập cho sinh viên khối ngành Kỹ thuật

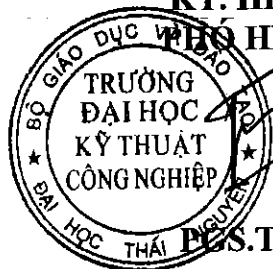
Ngày tháng năm 2023

Chủ nhiệm đề tài

(ký, họ tên)

Cơ quan chủ trì

KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG



TS. TS. Vũ Ngọc Pi

GIỚI THIỆU

1. Lý do chọn đề tài

Học phần Cấu trúc dữ liệu và giải thuật là học phần cơ sở được giảng dạy cho sinh viên khối ngành kỹ thuật. Đây là học phần sinh viên cần học tập, làm bài tập, xây dựng phần mềm trên máy tính vì vậy việc xây dựng video trực quan, sinh động, giúp sinh viên hiểu bài là rất cần thiết, giúp sinh viên có thể chủ động chuẩn bị bài học ở nhà, xem lại những nội dung chưa kịp hiểu trên lớp, tăng cường tính chủ động cho sinh viên.

Bên cạnh đó, giảng viên có thể giảm thời gian trình bày lý thuyết, tăng cường hướng dẫn thực hành, mở rộng số lượng bài tập cho sinh viên, giúp sinh viên hiểu bài hơn.

2. Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng video bài giảng cho học phần Cấu trúc dữ liệu và giải thuật

	3.2.2	Diễn giải và bảo vệ các kết quả thực hiện bài tập nộp, bài kiểm tra một cách khoa học, chuyên nghiệp	4
M3	4.1.3	Xây dựng chương trình từ thuật toán Phát triển chương trình phần mềm.	4

4. Mô tả tóm tắt học phần

Học phần trình bày các phương pháp tổ chức và những thao tác cơ sở trên từng cấu trúc dữ liệu, kết hợp với việc phát triển tư duy giải thuật để hình thành nên chương trình máy tính; Công cụ được sử dụng là các ngôn ngữ lập trình cấp cao như: Pascal, C, C++...; Các khái niệm: cấu trúc dữ liệu, giải thuật; Các phương pháp thiết kế giải thuật; Đánh giá độ phức tạp tính toán của giải thuật; Các giải thuật cơ bản như sắp xếp, tìm kiếm, đệ quy; Cấu trúc dữ liệu mảng và danh sách tuyến tính; Ngăn xếp và Hàng đợi; Cấu trúc cây; Đồ thị.

5. Nội dung và kế hoạch thực hiện học phần theo tuần

Tuần	Nội dung	CĐR học phần	Tài liệu học tập, tham khảo	Phương pháp dạy học
Chương 1. Mở đầu (3, 0, 6) (ghi chú: số tiết học trên lớp/số tiết thí nghiệm, thực hành/số tiết tự học)				Giảng, Thảo luận, Hướng dẫn bài tập
1-2	A. Nội dung giảng dạy - học tập Nội dung giảng dạy: - Giải thuật và cấu trúc dữ liệu - Cấu trúc dữ liệu và các vấn đề liên quan - Ngôn ngữ diễn đạt giải thuật Các cấu hình tổ hợp đơn giản. Nội dung học tập ở nhà: - Học lại kiến thức đã được học - Tìm hiểu thêm các thuật toán	1.2.1	[1],[2],[3]	
	B. Nội dung thực hành, thí nghiệm: (nếu không có ghi "không") Không			Giảng
Chương 2. Phân tích và thiết kế (3, 0, 6) (ghi chú: số tiết học trên lớp/số tiết thí nghiệm, thực hành/số tiết tự học)				
1-2	A. Nội dung giảng dạy - học tập Nội dung giảng dạy: - Modul hoá trong việc giải quyết bài toán - Tinh chỉnh từng bước - Phân tích giải thuật Nội dung học tập ở nhà: - Học lại kiến thức đã được học	1.2.1 2.1.1	[1],[2],[3]	Giảng
	B. Nội dung thực hành, thí nghiệm: (nếu			

	<p>A. Nội dung giảng dạy - học tập</p> <p>Nội dung giảng dạy:</p> <ul style="list-style-type: none"> - Các khái niệm cơ bản về cây - Cây tổng quát, cây nhị phân <p>Nội dung học tập ở nhà:</p> <ul style="list-style-type: none"> - Học lại kiến thức đã được học <p>B. Nội dung thực hành, thí nghiệm: (nếu không có ghi "không") Không</p>		[1],[2],[3]	Giảng
9-11		1.2.1 2.1.1 2.1.2		
Chương 7. Lý thuyết đồ thị (7, 0,14) (ghi chú: số tiết học trên lớp/số tiết thí nghiệm, thực hành/số tiết tự học)				
12-13	<p>A. Nội dung giảng dạy - học tập</p> <p>Nội dung giảng dạy:</p> <ul style="list-style-type: none"> - Định nghĩa và các khái niệm về đồ thị - Cây và cây khung của đồ thị <p>Nội dung học tập ở nhà:</p> <ul style="list-style-type: none"> - Học lại kiến thức đã được học <p>B. Nội dung thực hành, thí nghiệm: (nếu không có ghi "không") Không</p>	1.2.1 2.1.1 2.1.2	[1],[2],[3]	Giảng
Chương 8. Sắp xếp (3,0, 6)				
14-15	<p>A. Nội dung giảng dạy - học tập</p> <p>Nội dung giảng dạy:</p> <ul style="list-style-type: none"> - Một số phương pháp sắp xếp đơn giản - Sắp xếp kiểu phân đoạn (Quick sort) - Sắp xếp kiểu vun đống (Heap sort) - Sắp xếp kiểu hoà nhập <p>Nội dung học tập ở nhà:</p> <ul style="list-style-type: none"> - Học lại kiến thức đã được học <p>B. Nội dung thực hành, thí nghiệm: (nếu không có ghi "không") Không</p>	1.2.1 2.1.1 2.1.2	[1],[2],[3]	Giảng
Chương 9. Tìm kiếm (3, 0, 6)				
14-15	<p>A. Nội dung giảng dạy - học tập</p> <p>Nội dung giảng dạy:</p> <ul style="list-style-type: none"> - Tìm kiếm tuần tự - Tìm kiếm nhị phân - Cây nhị phân tìm kiếm 	1.2.1 2.1.1 2.1.2	[1],[2],[3]	Giảng

		bản. - Vận dụng các ngôn ngữ lập trình đã học để triển khai thuật toán thành chương trình.	
	Phân tích	Phân tích hoạt động và kết quả của các thuật toán và chương trình.	15
3	Đánh giá	Nhận xét và đưa ra kết luận về hiệu quả, độ phức tạp của các thuật toán và chương trình.	10
	Sáng tạo	Đề xuất thuật toán cải tiến cho bài toán cụ thể	5

Ghi chú: Nội dung này nhằm phục vụ xây dựng câu hỏi kiểm tra quá trình, Ngân hàng câu hỏi thi, đề thi kết thúc học phần và đánh giá kết quả kiểm tra hoặc thi.

8. Tài liệu học tập

8.1. Sách, giáo trình chính:

1. Đỗ Xuân Lôi; Cấu trúc dữ liệu và giải thuật; NXB Khoa học kỹ thuật 1996

8.2. Sách tham khảo:

1. Nguyễn Thị Hương, Giáo trình cấu trúc dữ liệu và giải thuật; NXB Khoa học kỹ thuật 2010
2. Phạm Văn Át, C++ và lập trình hướng đối tượng, NXB Khoa học và kỹ thuật 1999.

9. Phụ trách học phần

- Bộ môn: Tin học Công nghiệp

- Giảng viên giảng dạy chính: *(Yêu cầu mỗi HP phải bố trí tối thiểu từ 02 giảng viên giảng dạy chính).*

1. ThS. Nguyễn Thị Hương Email: huongktpm@tnut.edu.vn
2. ThS. Đỗ Duy Cốp Email: duycop@tnut.edu.vn
1. ThS. Trần Thị Ngọc Linh Email: tranngoclinh@tnut.edu.vn

10. Phê duyệt

Trưởng khoa

Trưởng Bộ môn

Đại diện nhóm Biên soạn

PGS. TS. Đào Huy Du

TS. Nguyễn Tuấn Linh

ThS. Nguyễn Thị Hương

3. ThS. Đặng Thị Hiền Email: dangthihien@tnut.edu.vn

4. ThS. Trần Thị Ngọc Linh Email: tranngoclinh@tnut.edu.vn

5. ThS. Trần Thị Thanh Email: tranthithanh@tnut.edu.vn

6. ThS. Đỗ Duy Cốp Email: duycop@tnut.edu.vn

1.2. Đối tượng sử dụng

Video bài giảng sẽ được sử dụng làm tài liệu giảng dạy cho các Thầy, cô của bộ môn và làm tài liệu học tập cho toàn bộ sinh viên khối ngành kỹ thuật.

CHƯƠNG 2: TRIỂN KHAI THỰC HIỆN

2.1. Phương pháp triển khai

• **Số lượng video:** Đối với học phần Cấu trúc dữ liệu và giải thuật, việc xây dựng video được bám sát theo yêu cầu của đề cương môn học. Tuy nhiên, mỗi video không xây dựng theo toàn bộ tiết học, mà lựa chọn những nội dung trọng tâm nhất để xây dựng video. Số lượng video của học phần này là 24 video có độ dài từ 08 – 20 phút.

• **Bố cục video:** Mỗi video sẽ diễn giải về một nội dung riêng tương ứng với các nội dung trọng tâm đã được lựa chọn. Trong mỗi video bao gồm 3 phần chính:

- **Lý thuyết:** Giảng viên giảng bài trên file PowerPoint
- **Bài tập về nhà:** bên cạnh việc hiểu được lý thuyết, biết cách viết mã lệnh theo hướng dẫn của giảng viên, sinh viên còn cần biết ứng dụng các nội dung đã học để làm bài tập. Các bài tập về nhà được đưa ra sát với nội dung học, giúp sinh viên hiểu hơn về những nội dung đã học.

• **Nội dung video:** Bám sát đề cương môn học

2.2. Công cụ thực hiện

Để triển khai xây dựng đề tài

- Bài giảng được xây dựng trên PoperPoint 2010, bao gồm toàn bộ nội dung giảng dạy của học phần.

- Việc chỉnh sửa nội dung video được thực hiện qua Video Editor: Đây là phần mềm được cung cấp sẵn bởi Windows, khá dễ sử dụng và đầy đủ chức năng để chỉnh sửa video.

2.3. Phân bổ số lượng video

Nội dung đề cương chi tiết	Tên video BG	Nội dung video	Thời lượng (phút)	Đánh giá
Chương 1. MỞ ĐẦU	Bai giảng Video_Chuong1_CTDL&G T_Nguyen Thi Huong	Chương I: MỞ ĐẦU 1. Giải thuật và cấu trúc dữ liệu 2. Cấu trúc dữ liệu và các vấn đề liên quan 3. Ngôn ngữ diễn đạt giải thuật	10p	
Chương 2. THIẾT KẾ VÀ PHÂN	Bai giảng Video_Chuong2.1_CTDL >_Nguyen Thi Huong Bai giảng	Chương II: THIẾT KẾ VÀ PHÂN TÍCH 1. Từ bài toán đến chương trình	7-15 phút mỗi video	

	Bai giang Video_Chuong6.5_CTDL >_Nguyen Thi Huong			
Chương 7. LÝ THUYẾT ĐỒ THỊ	Bai giang Video_Chuong7.1_CTDL >_Nguyen Thi Huong Bai giang Video_Chuong7.2_CTDL >_Nguyen Thi Huong Bai giang Video_Chuong7.3_CTDL >_Nguyen Thi Huong Bai giang Video_Chuong7.4_CTDL >_Nguyen Thi Huong	Chương VII: LÝ THUYẾT ĐỒ THỊ 1. Định nghĩa và các khái niệm về đồ thị 2. Đồ thị vô hướng, đồ thị có hướng 3. Đường đi, chu trình. Đồ thị liên thông 4. Cây và cây khung của đồ thị	8-20 phút mỗi video	
Chương 8. SẮP XẾP	Bai giang video_chuong 8.1_CTDL>_Nguyen Thi Huong Bai giang video_chuong 8.2_CTDL>_Nguyen Thi Huong Bai giang video_chuong 8.3_CTDL>_Nguyen Thi Huong	Chương VIII: SẮP XẾP 1. Thuật toán lựa chọn “Selection sort” 2. Thuật toán chèn “Insertion sort” 3. Thuật toán nổi bọt “Bubble sort” 4. Thuật toán sắp xếp vun đống “Heap sort” 5. Thuật toán sắp xếp nhANH “Quick sort”	8-20 phút mỗi video	
Chương 9. TÌM KIẾM	Bai giang Video_Chuong9_CTDL&G T_Nguyen Thi Huong	Chương IX: TÌM KIẾM 1. Đặt vấn đề 2. Tìm kiếm tuần tự 3. Tìm kiếm nhị phân	10p	
Tổng số	24 video			

CHƯƠNG 4: KẾT LUẬN

4.1. Kết luận

Đề tài xây dựng video bài giảng cho học phần Cấu trúc dữ liệu là một đề tài triển khai theo đặt hàng của nhà trường. Đề tài này có tính ứng dụng cao, là một tài liệu học tập tốt cho sinh viên. Giúp nâng cao tính tự giác, tự nghiên cứu của sinh viên.

Tác giả mong rằng sẽ nhận được nhiều hơn nữa sự đánh giá khách quan, thực tế từ phía hội đồng đánh giá, các đồng nghiệp để nâng cao khả năng chuyên môn và hoàn thiện đề tài.

4.2. Hướng phát triển của đề tài

Hoàn thiện, chỉnh sửa chất lượng video cho tốt hơn. Bổ sung thêm các nội dung về Word, Excel, PowerPoint, ngoài ra cũng bổ sung thêm video hướng dẫn các bài tập mở rộng cho sinh viên.



BÀI GIẢNG

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

- Hai yếu tố tạo nên một chương trình máy tính
 - Cấu trúc dữ liệu
 - Giải thuật

Cấu trúc dữ liệu + Giải thuật = Chương trình



MỤC TIÊU

Trang bị cho sinh viên các kiến thức về

Tư duy xây dựng các giải thuật và lập trình

Các cấu trúc thuật toán (giải thuật) cơ bản



CHƯƠNG 1: MỞ ĐẦU

1.1 Giải thuật và cấu trúc dữ liệu

Giải thuật (statements)

Giải thuật chỉ phản ánh các phép xử lý:



CHƯƠNG 1: MỞ ĐẦU

1.2 Các tính chất của thuật toán

- **Nhập (input):** Các thuật giải có các giá trị nhập (input values) từ một tập hợp nhất định nào đó.
- **Xuất (output):** Từ mỗi tập hợp các giá trị được nhập một thuật toán thường tạo ra những giá trị xuất (output values) thuộc một tập hợp nhất định nào đó thể hiện lời giải cho bài toán.
- **Tính xác định (definiteness):** Các bước trong thuật toán phải chính xác rõ ràng.





CHƯƠNG 1: MỞ ĐẦU

Ví dụ : tìm phần tử lớn nhất trong mảng A

Giải thuật tìm $\text{Max}(A, n)$

Input: Mảng A , gồm n số nguyên

Output: Giá trị lớn nhất của A

$Max \leftarrow A[0]$

for $i \leftarrow 1$ to $n - 1$ do

 if $A[i] > Max$ then

$Max \leftarrow A[i]$

return Max



Diễn đạt giải thuật

Các nút biểu diễn giải thuật bằng sơ đồ khối



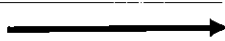
Nút thao tác:



Nút điều khiển: trong đó ghi điều kiện cần kiểm tra trong quá trình tính toán.



Nút khởi đầu, kết thúc:



Cung :





CHƯƠNG 1: MỞ ĐẦU

1. $\Delta = b^2 - 4ac$

2. if $\Delta > 0$ then

begin

$x_1 := \frac{-b - \sqrt{\Delta}}{2a};$

$x_2 := \frac{-b + \sqrt{\Delta}}{2a};$



CHƯƠNG 1: MỞ ĐẦU

Xuất kết quả: phương trình có hai nghiệm là x_1 và x_2 ;

end

3. else if $\Delta = 0$ then

Xuất kết quả: phương trình có nghiệm kép là $-\frac{b}{2a}$

4. else { trường hợp $\Delta < 0$ }

Xuất kết quả: phương trình vô nghiệm;





CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

THIẾT KẾ GIẢI THUẬT

Module hoá và việc giải quyết bài toán

- Chiến thuật chia để trị (divide-conquer):
- Để thực hiện chiến thuật này, thường có hai cách thiết kế:
 1. Từ trên xuống (Top-Down Design).
 2. Tinh chỉnh từng bước



CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

Ví dụ: Bài toán sắp xếp một dãy n số, theo thứ tự tăng dần

- Chọn số bé nhất trong n số để vào vị trí thứ 1
- Chọn số bé nhất trong $n-1$ số còn lại để vào vị trí thứ 2
-
- Chọn số bé nhất trong 2 số còn lại để vào vị trí thứ $n-1$





CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

PHÂN TÍCH GIẢI THUẬT

Khi một giải thuật được xây dựng, hàng loạt yêu cầu đặt ra

- Yêu cầu về tính đúng đắn của giải thuật
- Tính đơn giản của giải thuật.
- Yêu cầu về không gian :
- Yêu cầu về thời gian :



CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

PHÂN TÍCH GIẢI THUẬT

Giải quyết bài toán

- Phải đứng trước việc lựa chọn giải thuật nào ?
- Dựa trên cơ sở nào để lựa chọn ?





CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

Độ phức tạp của thuật toán

- *Độ phức tạp của thuật toán* được thể hiện qua khối lượng thời gian và không gian để thực hiện thuật toán.

Không gian ở đây được hiểu là các yêu cầu về bộ nhớ, thiết bị lưu trữ, ... của máy tính để thuật toán có thể làm việc được

Trong phần này chúng ta chỉ đề cập đến độ phức tạp về thời gian của thuật toán.



CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

Độ phức tạp của thuật toán

- Độ phức tạp thời gian của thuật toán dựa vào số lượng thao tác được sử dụng trong thuật toán và số lượng thao tác này phụ thuộc vào cỡ (size) của dữ liệu nhập.
- Ta còn gọi độ phức tạp thời gian của thuật toán là *độ phức tạp tính toán*.



CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

ĐỘ PHỨC TẠP CỦA THUẬT TOÁN

Trong thuật toán nếu ta xem thời gian thực hiện thuật toán là số lần thực hiện phép so sánh hay phép gán thì thời gian thực hiện thuật toán trong trường hợp xấu nhất là:

$$t(n) = 1 + 2*(n-1)$$



CHƯƠNG 2: THIẾT KẾ VÀ PHÂN TÍCH

ĐỘ PHỨC TẠP CỦA THUẬT TOÁN

Thời gian thực hiện thuật toán trong trường hợp tốt nhất là:

$$T(n) = 1 + (n-1) = n.$$

Ký hiệu O

Ví dụ 2: Đánh giá độ phức tạp (thời gian) của thuật toán tìm kiếm tuyến tính

Thời gian thực hiện thuật toán trong trường hợp tốt nhất là: $O(1)$.



CHƯƠNG 3: ĐỆ QUY VÀ GIẢI THUẬT ĐỆ QUY

Thuật toán cần phải thỏa mãn 3 tính chất

- Tính hữu hạn.
- Tính xác định
- Tính đúng đắn



CHƯƠNG 3: ĐỆ QUY VÀ GIẢI THUẬT ĐỆ QUY

Nội dung:

Khái niệm đệ quy

Giải thuật và chương trình đệ quy

Thiết kế giải thuật đệ quy

Ưu nhược điểm của đệ quy

Một số dạng giải thuật đệ quy





CHƯƠNG 3: ĐỆ QUY VA GIAI THUAT ĐỆ QUY

Giải thuật và hàm đệ quy

- Giải thuật đệ quy

Nếu bài toán T được thực hiện bằng lời giải của bài toán T' có dạng giống T là lời giải đệ quy

Giải thuật tương ứng với lời giải như vậy gọi là giải thuật đệ quy.

- Hàm đệ quy



CHƯƠNG 3: ĐỆ QUY VA GIAI THUAT ĐỆ QUY

Giải thuật đệ quy

Ví dụ: Xét bài toán tìm một từ trong quyển từ điển:

```
if (từ điển là một trang)
    tìm từ trong trang này
else {
    Mở từ điển vào trang "giữa"
    Xác định xem nửa nào của từ điển chứa từ cần
    tìm;
    if (từ đó nằm ở nửa trước)
        tìm từ đó ở nửa trước
    else tìm từ đó ở nửa sau.
}
```





CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Ví dụ 1 : Hàm Fact(n) tính số hạng n của dãy n!, định nghĩa như sau:

$$\text{fact}_0 = 1 ;$$

$$\text{fact}_n = n * \text{fact}_{n-1}; (n \geq 1)$$

```
long int Fact(int n)
```

```
{
```

```
  if (n == 0)
```

```
    return 1;
```

```
  else
```

```
    return n * Fact(n-1);
```



CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Ví dụ 2: Tính số hạng thứ n của dãy Fibonacci được định nghĩa như sau:

$$f_1 = f_0 = 1 ;$$

$$f_n = f_{n-1} + f_{n-2}; (n > 1)$$

```
int Fibo(int n)
```

```
{
```

```
  if ( n < 2 )
```

```
    return 1 ;
```

```
  else
```

```
    return (Fibo(n - 1) + Fibo(n - 2)) ;
```

```
}
```





CHƯƠNG 3: ĐỆ QUY VA GIẢI THUẬT ĐỆ QUY

Ưu điểm của đệ quy

- Sáng sủa, dễ hiểu, nêu rõ bản chất vấn đề
- Tiết kiệm thời gian hiện thực mã nguồn

Nhược điểm của đệ quy

- Tốn nhiều bộ nhớ, thời gian thực thi lâu
- Một số bài toán không có lời giải đệ quy



CHƯƠNG 3: ĐỆ QUY VA GIẢI THUẬT ĐỆ QUY

**Một số bài toán giải
bằng giải thuật đệ quy
điển hình**



CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Bài toán tháp Hà Nội

- Trường hợp $n = 1$

Chuyển từ A sang C

- Trường hợp $n > 1$

Chuyển $(n-1)$ đĩa từ A sang B, C
trung gian

Chuyển đĩa n từ A sang C

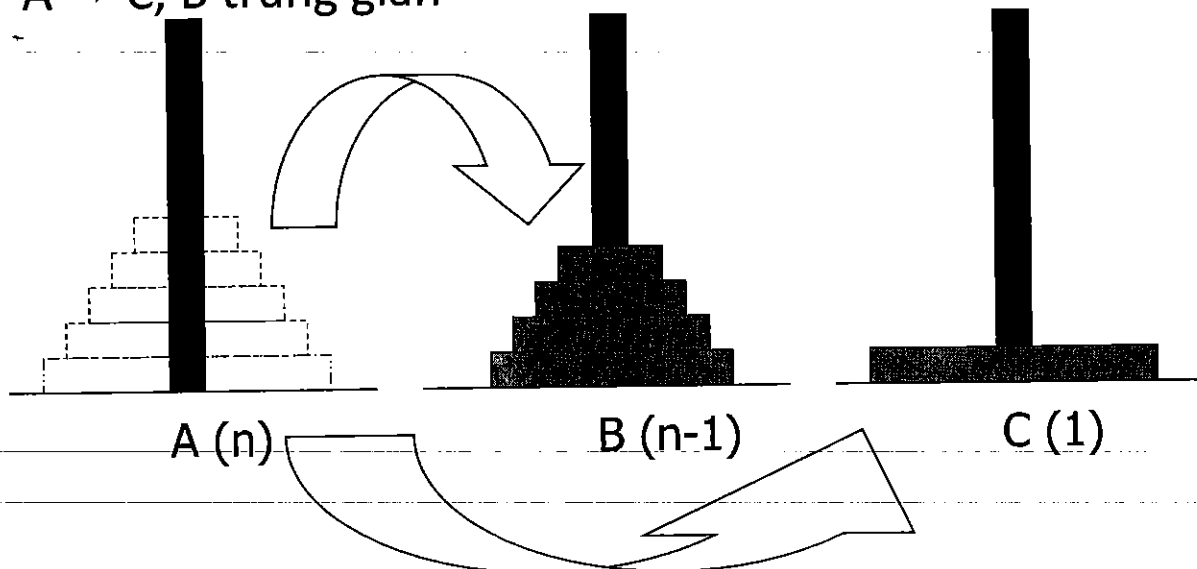
Chuyển $(n-1)$ đĩa từ B sang C, A
làm trung gian



CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Bài toán tháp Hà Nội

A \rightarrow C, B trung gian

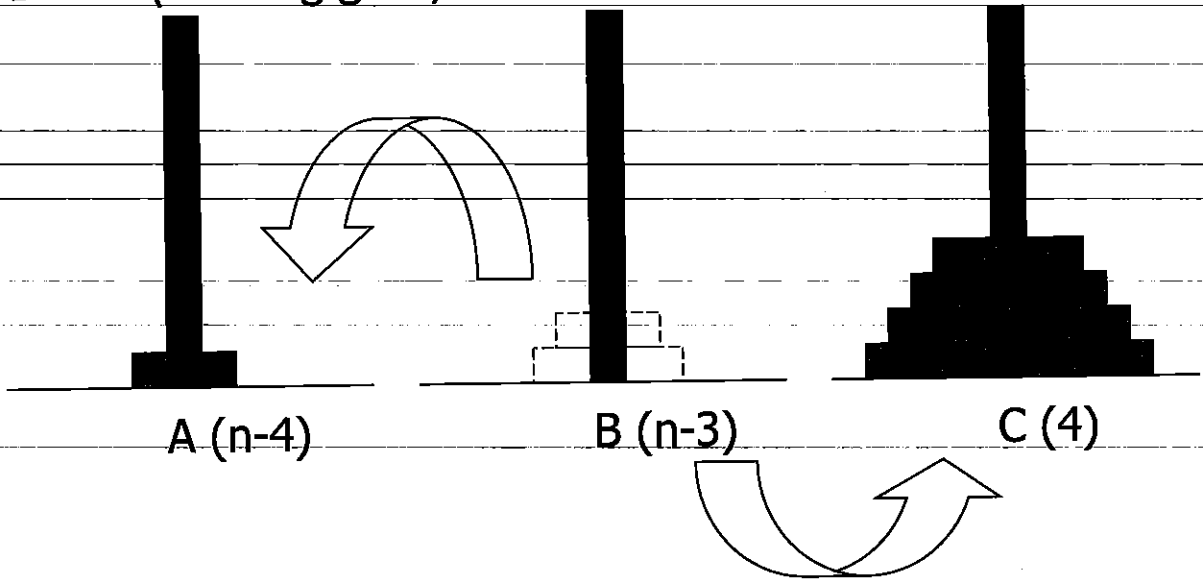




CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Bài toán tháp Hà Nội

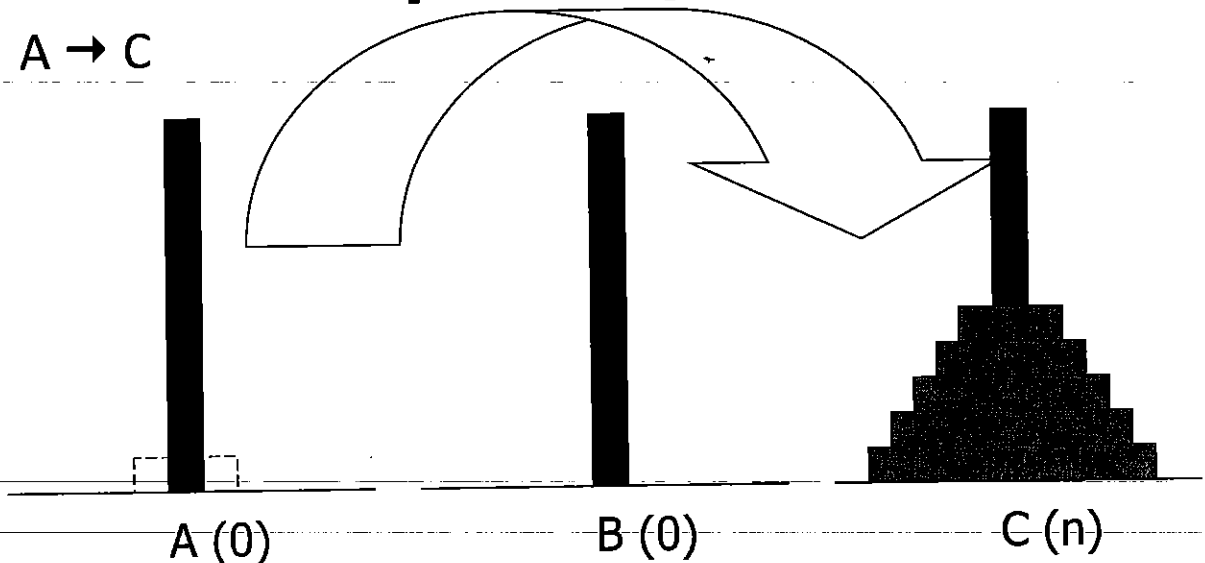
B → C (A trung gian)



CHƯƠNG 3: ĐỆ QUI VA GIẢI THUẬT ĐỆ QUI

Bài toán tháp Hà Nội

A → C





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

4.1 Các khái niệm

Mảng: Là một tập hợp có thứ tự gồm một số cố định các phần tử

Mỗi phần tử của mảng ngoài giá trị (info) còn được đặc trưng bởi chỉ số để biểu hiện thứ tự của nó trong mảng. Có mảng 1 chiều, 2, 3 chiều...n chiều.



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Danh sách:

Là một tập có thứ tự nhưng bao gồm một số biến động các phần tử. Phép bổ sung và loại bỏ một phần tử là phép thường xuyên tác động lên danh sách. Một danh sách mà quan hệ lân cận giữa các phần tử được hiển thị ra gọi là danh sách tuyến tính.





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

4.2 Cấu trúc lưu trữ của mảng

Một véc tơ A có n phần tử, nếu mỗi phần tử $a[i]$ ($1 \leq i \leq n$) chiếm c từ máy thì nó sẽ được lưu trữ trong cn từ máy kế tiếp nhau (lưu trữ kế tiếp – Sequential storage allocation)



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

L_0 : địa chỉ gốc (là địa chỉ của từ máy đầu tiên trong miền nhớ kế tiếp dùng để lưu trữ véc tơ)

Địa chỉ ai sẽ được tính bởi:

$$\text{Loc}(a_i) = L_0 + c * (i-1)$$

$F(i) = c * (i-1)$ gọi là hàm địa chỉ (addressfunction)



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Tổng quát:

Đối với ma trận n hàng, m cột →

$$\text{Loc}(a_{ij}) = L0 + (j-1) * n + (i-1)$$



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Giả sử xét bài toán tính đa thức của x, y:
chẳng hạn cộng 2 đa thức (hay trừ, nhân
chia....)



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Nhược điểm:

Số mũ của đa thức bị hạn chế bởi cấp của ma trận \rightarrow các đa thức được xử lý bị giới hạn trong phạm vi hẹp

Ma trận biểu diễn nhiều phần tử 0 gây ra sự lãng phí bộ nhớ. Gọi là ma trận thưa.



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

4.3 Lưu trữ kế tiếp của d/s tuyến tính

Do số phần tử của d/s tuyến tính thường biến động nên việc lưu trữ chỉ đảm bảo được nếu biết được $m = \max(n)$, (n - kích thước d/s). Gây lãng phí bộ nhớ vì có hiện tượng "Giữ chỗ để đầy" mà không dùng tới





CHƯƠNG 4 MẢNG VÀ DANH SÁCH

T: con trỏ, trỏ tới đỉnh Stack, biến đổi khi stack hoạt động

T giảm 1: Khi loại bỏ 1 phần tử ra khỏi stack

T tăng 1 : Khi thêm một phần tử vào stack

T=0 : Khi stack rỗng

Const MaxStack=?

Type Item=?



CHƯƠNG 4 MẢNG VÀ DANH SÁCH

4.4.2 Lưu trữ Stack bằng mảng (lưu trữ kế tiếp)

Có thể lưu trữ Stack bằng một véc tơ lưu trữ S, gồm n phần tử nhớ kế tiếp

Stack=array[0..MaxStack] of item;

Top=0;

Top= MaxStack;



CHƯƠNG 4 MẢNG VÀ DANH SÁCH

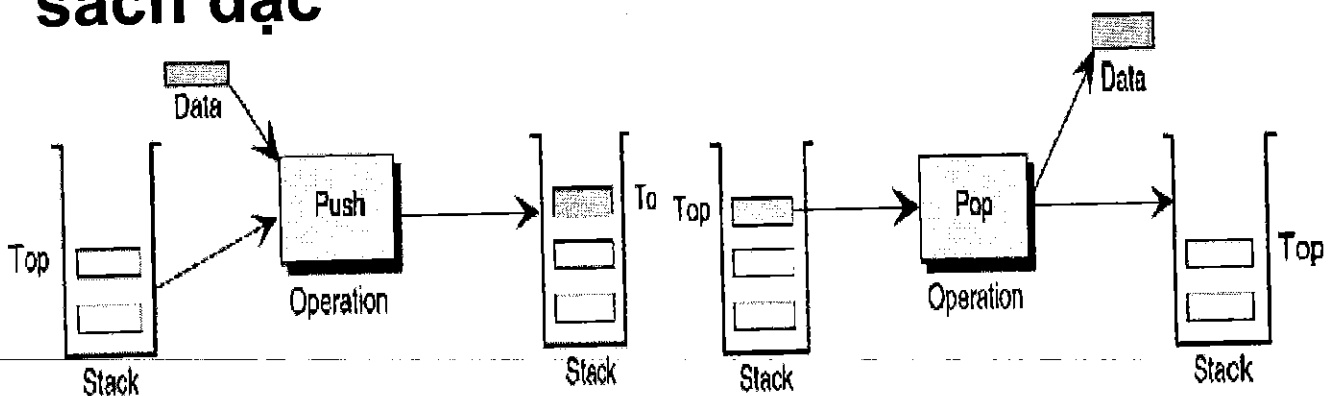
Các thao tác cơ bản trên Stack:

- Stack: khởi tạo Stack rỗng
- isEmpty: kiểm tra Stack rỗng ?
- isFull: kiểm tra Stack đầy ?
- Push: thêm 1 phần tử vào đỉnh Stack, có thể làm Stack đầy
- Pop: lấy ra 1 phần tử từ đỉnh Stack, có thể làm Stack rỗng



CHƯƠNG 4 MẢNG VÀ DANH SÁCH

Tổ chức Stack theo kiểu danh sách đặc



Thao tác Push

Thao tác Pop





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Kiểm tra Stack đầy

```
boolean IsFull()  
{  
    if (StkTop == StkMax-1)  
        return true; // Stack đầy  
    return false; // Stack chưa đầy  
}
```



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Thêm một phần tử vào Stack

```
boolean Push(int newItem)  
{  
    if (IsFull())  
        return false;  
    // Stack đầy, không thêm vào được  
    StkTop++;  
    StkArr[StkTop] = newItem;  
    return true; // Thêm thành công  
}
```





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Queue là 1 cấu trúc dữ liệu:

Gồm nhiều phần tử có thứ tự

Hoạt động theo cơ chế “Vào trước – Ra trước” (FIFO – First In, First Out)



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Cấu trúc của Queue

- Dùng 1 mảng (QArray) để chứa các phần tử
- Dùng 1 số nguyên (QMax) để lưu số phần tử tối đa trong hàng đợi
- Dùng 2 số nguyên (QFront, QRear) để xác định vị trí Đầu, Cuối hàng đợi
- Dùng 1 số nguyên (QNumItems) để lưu số phần tử hiện có trong hàng đợi





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Khai báo lớp Queue

// Giả sử Queue chứa các phần tử kiểu nguyên (int)

```
Class Queue {  
    private int [] QArray;  
    private int QMax;  
    private int QNumItems;  
    private int QFront;  
    private int QRear;
```



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Khai báo lớp Queue

// Khởi tạo Queue chứa các phần tử kiểu nguyên (int)

```
public Queue(int size) {  
    QArray = new int[size];  
    QMax = size;  
    QFront = QRear= -1; // Queue rỗng  
    QNumItems = 0; // chưa có phần tử  
    nào trong Queue
```





CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Thêm 1 phần tử vào Queue

- Thao tác: thêm 1 phần tử vào cuối Queue
boolean Append(int newItem)

```
{
    if (IsFull()) return false;
    //Queue đầy, không thêm vào được
    QRear++;
    QArray[q.QRear] = newItem;
    // thêm phần tử vào cuối Queue
    QNumItems++;
    return true; // Thêm thành công
}
```



CHƯƠNG 4

MẢNG VÀ DANH SÁCH

Lấy một phần tử ra khỏi Queue

- Thao tác: lấy ra 1 phần tử ở đầu Queue

boolean Take(int itemout)

```
{if (IsEmpty()) return false; // Queue rỗng, không lấy ra
    được
    itemout = QArray[QFront]; // lấy phần tử đầu ra
    QFront++;
    QNumItems--;
    if (QFront==QMax) // nếu đi hết mảng ...
    QFront = QRear = -1 ; // ... quay trở về đầu mảng
    return true; // lấy thành công
}
```



**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

Danh sách móc nối cũng gồm nhiều phần tử như mảng, nhưng có sự khác biệt là các phần tử không nằm liên tiếp nhau trong bộ nhớ, mà dùng liên kết để móc nối với nhau.

**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

Khái niệm về danh sách móc nối

-Danh sách móc nối (linked list) là một cấu trúc dữ liệu bao gồm một nhóm các nút (nodes) tạo thành một chuỗi. Thông thường mỗi nút gồm dữ liệu (data) ở nút đó và tham chiếu (reference) đến nút kế tiếp trong chuỗi.



**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

Phân loại danh sách móc nối

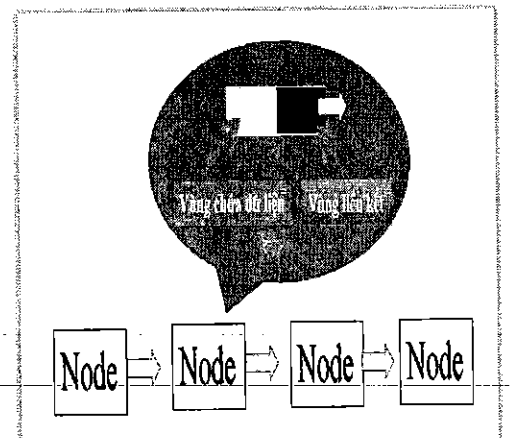
1. Danh sách móc nối đơn (single-link list)
2. Danh sách móc nối kép (double - lists)
3. Danh sách móc nối vòng (Circulary Linked List)

**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)****5.1. DANH SÁCH MÓC NỐI ĐƠN**

▪ Danh sách móc nối đơn là danh sách có nhiều nút móc nối đơn lại với nhau

▪ Nút của danh sách gồm:

1. Một số từ máy kế tiếp nhau
2. Nút có thể nằm ở vị trí bất kỳ trong bộ nhớ
3. Mỗi nút chứa những thông tin ứng với một phần tử và địa chỉ phần tử đứng sau



**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

❖ Nếu danh sách rỗng thì cho L bằng nút đó.
Nếu không thì thêm nút đó vào cuối/đầu

```
node *tg=taonut(a);
    if (L==NULL) L=tg;
    else insert_cuoi(tg); // else
insert_dau(tg);
```

**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

❖ Thêm một nút vào cuối danh sách

```
void insert_cuoi(node *p)// them mot nut p vao cuoi
danh sach
```

```
{
    node *tg;
    tg=L;
    while (tg->link!=NULL) tg=tg->link;
    tg->link=p;
}
```





CHƯƠNG 5

DANH SÁCH MÓC NỐI (link list)

❖ Xóa một nút ở đầu danh sách

```
void delete_dau()// xoa mot nut o dau danh sach
{
    if (L==NULL) cout<<"Danh sach rong";
    else
    {
        L=L->link;
    }
}
```



CHƯƠNG 5

DANH SÁCH MÓC NỐI (link list)

Chú ý:

Chỉ có phần tử đầu tiên được truy cập trực tiếp, các phần tử khác chỉ được truy nhập sau khi truy nhập phần tử trước nó

Tồn bộ nhớ vì có thêm trường LINK ở mỗi nút để lưu trữ địa chỉ tiếp theo



**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

```
1. Tạo một nút mới có giá trị x
node *taonut(int x)//tao nut co gia tri x
{
    node *pp=new node();
    pp->info=x;
    pp->lptr=NULL;
    pp->rptr=NULL;
    return pp;
}
```

**CHƯƠNG 5****DANH SÁCH
MÓC NỐI (link list)**

2. Bổ sung một nút vào cuối/đầu danh sách
❖ Nếu danh sách rỗng thì cho L bằng nút đó. Nếu không thì thêm nút đó vào cuối/đầu

```
node *tg=taonut(a);
if (R==NULL)
    {L=tg;
    R=tg;}
else insert_cuoi(tg); //else
insert_dau(tg);
```





CHƯƠNG 5

**DANH SÁCH
MÓC NỐI (link list)****3. Loại bỏ một nút ra khỏi danh sách nối kép.****❖ Xóa một nút ở cuối danh sách**

```
void delete_cuoi()//xoa mot nut o cuoi danh sach
```

```
{  
    node *tg;  
    if (R==NULL) cout<<"Danh sach rong";  
    else  
    {  
        R=R->lptr;  
        delete(R->rptr);  
        R->rptr=NULL;  
    }  
}
```



CHƯƠNG 5

**DANH SÁCH
MÓC NỐI (link list)****❖ Xóa một nút ở đầu danh sách**

```
void delete_dau()// xoa mot nut o dau danh sach
```

```
{  
    if (R==NULL) cout<<"Danh sach rong";  
    else  
    {  
        L=L->rptr;  
        delete(L->lptr);  
        L->lptr=NULL;  
    }  
}
```





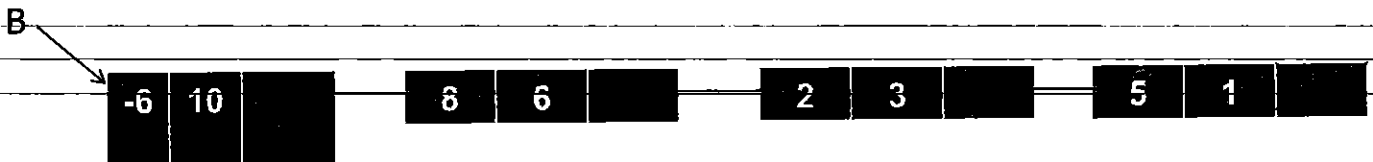
Như vậy đa thức:

$A(x) = 4x^{10} - 2x^3 + 6$ sẽ được biểu diễn bởi:



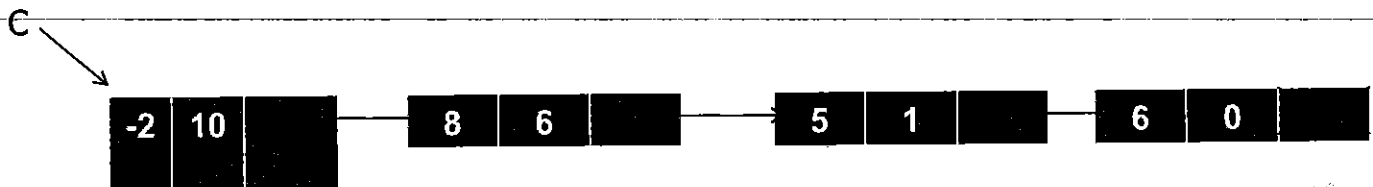
Còn đa thức:

$B(x) = -6x^{10} + 8x^6 + 2x^3 + 5x$, sẽ có dạng



Phép cộng hai đa thức này sẽ cho:

$$C(x) = A(x) + B(x) = -2x^{10} + 8x^6 + 5x + 6$$



Trước hết cần phải thấy rằng để thực hiện $A(x)$ với $B(x)$ ta phải tìm đến từng số hạng của đa thức đó, nghĩa là phải dùng hai biến trỏ p và q để duyệt qua hai danh sách tương ứng với $A(x)$ và $B(x)$ trong quá trình tìm này.

❖ Ta sẽ thấy có những tình huống sau:

a, $EXP(p) = EXP(q)$ ta sẽ phải thực hiện cộng giá trị COEF ở hai nút đó, nếu giá trị tổng khác không thì phải tạo ra nút mới thể hiện số hạng tổng đó và gắn vào danh sách ứng với $C(x)$.

b, Nếu $EXP(p) > EXP(q)$ (hoặc ngược lại thì cũng tương tự): phải sao chép nút p và gắn vào danh sách của $C(x)$.

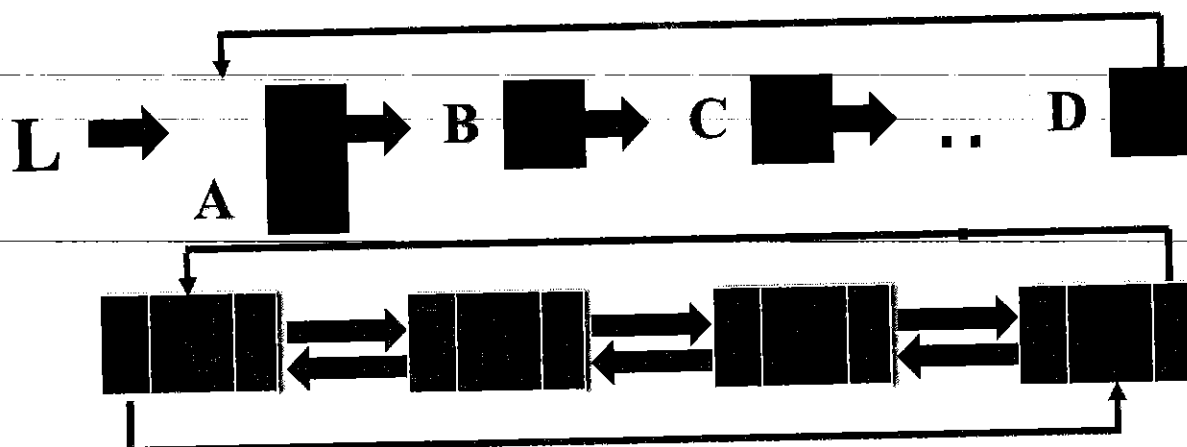
c, Nếu một danh sách kết thúc trước phần còn lại của danh sách kia sẽ được sao chép và gắn dần vào danh sách của $C(x)$.

❖ Mỗi lần một nút mới được tạo ra đều phải gắn vào "đuôi" của C . Như vậy phải thường xuyên nắm được nút đuôi này bằng một con trỏ d .



5.4. DANH SÁCH MÓC NỐI VÒNG

- Cả hai loại Danh sách liên kết đơn (Singly Linked List) và danh sách liên kết kép (Doubly Linked List) đều có thể được tạo thành dạng danh sách liên kết vòng.
- Hình ảnh minh họa cho móc nối vòng



5.4. DANH SÁCH MÓC NỐI VÒNG

❖ Ưu điểm:

- Giúp cho việc truy nhập vào các node được linh hoạt hơn (vì node nào cũng có thể coi là node đầu tiên và con trỏ L trở tới node nào cũng được)
- Phép ghép, tách cũng có những thuận lợi nhất định





5.4. DANH SÁCH MỐC NỐI VÒNG

❖ Sau đây là đoạn giải thuật bổ sung một nút vào thành nút đầu tiên của một danh sách nối vòng có “nút đầu danh sách” trở bởi HEAD

```
node *pp=new node()
pp->info=x;
pp->link=HEAD->link;
HEAD->link=pp;
```

10/11/2023

12
9

5.5 STACK VÀ QUEUE MỐC NỐI

- ❖ Như ta đã biết đối với stack việc truy nhập đều thực hiện ở một đầu (đỉnh).
- ❖ Chẳng hạn với danh sách nối đơn trở bởi L thì có thể coi L như con trỏ trở tới đỉnh stack.
- ❖ Bổ sung một nút vào stack chính là bổ sung một nút vào thành nút đầu tiên của danh sách; loại bỏ một nút ra khỏi stack chính là loại bỏ nút đầu tiên của danh sách đang trở bởi L.





5.5 STACK VÀ QUEUE MỐC NỔI

- Dùng con trỏ đòi hỏi bộ nhớ phụ (tốn bộ nhớ).
- Danh sách liên kết không cho phép truy cập trực tiếp.
- Tốn thời gian cho việc duyệt và biến đổi con trỏ.



CHƯƠNG 6

CÂY

Mục tiêu

Trang bị cho sinh viên các khái niệm và ứng dụng cây

Cài đặt và thực hiện các phép toán trên cây, đặc biệt là các phép toán trên cây nhị phân tìm kiếm.



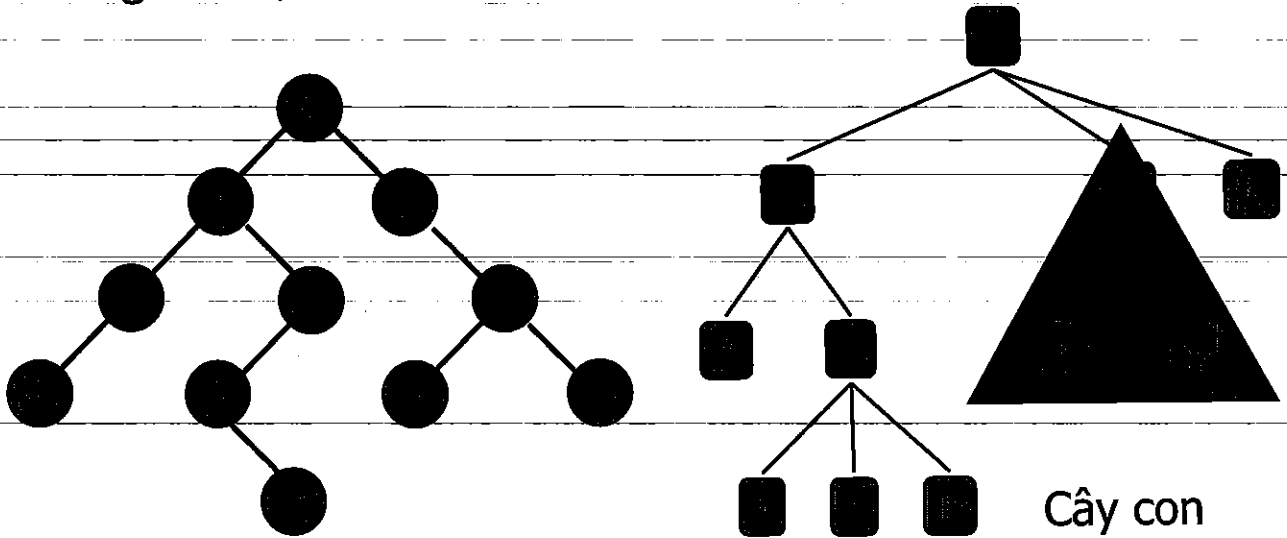


CHƯƠNG 6

CÂY

Biểu diễn cây

Bằng đồ thị



Cây con

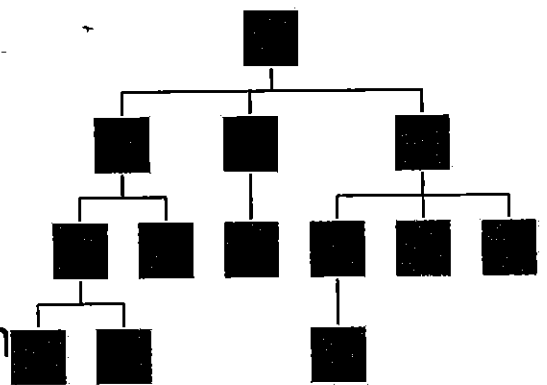


CHƯƠNG 6

CÂY

Các thuật ngữ

- Bậc của nút và bậc của cây
Nút A: bậc 3, nút C bậc 1
Bậc của cây: 3
- Nút gốc, Nút lá và nút nhánh
- Nút cha (Parent), nút con (children)



**CHƯƠNG 6****CÂY**

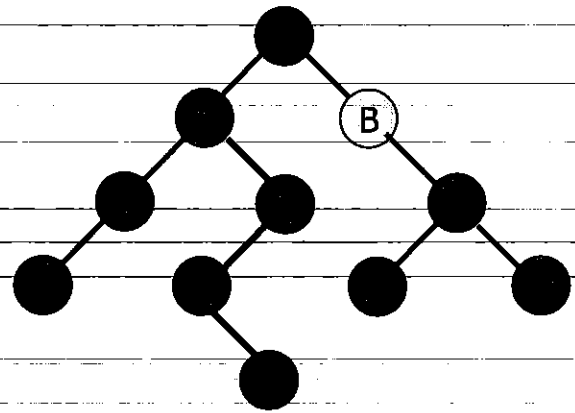
- Tổ tiên (ancestors) của một nút

Tổ tiên của nút J

- Con cháu (Descendant) của một nút:

Con cháu của B

- Các con của cùng một cha gọi là anh em ruột (siblings)

**CHƯƠNG 6****CÂY****Cây có thứ tự và Rừng**

- Cây có thứ tự (ordered tree)
 - Một cây gọi là có thứ tự khi ta thay đổi vị trí của các cây con, ta nhận được một cây mới
- Rừng (forest)
 - Tập hợp hữu hạn các cây phân biệt
 - Nếu bỏ đi nút gốc của một cây, ta sẽ thu được một rừng gồm nhiều cây phân biệt





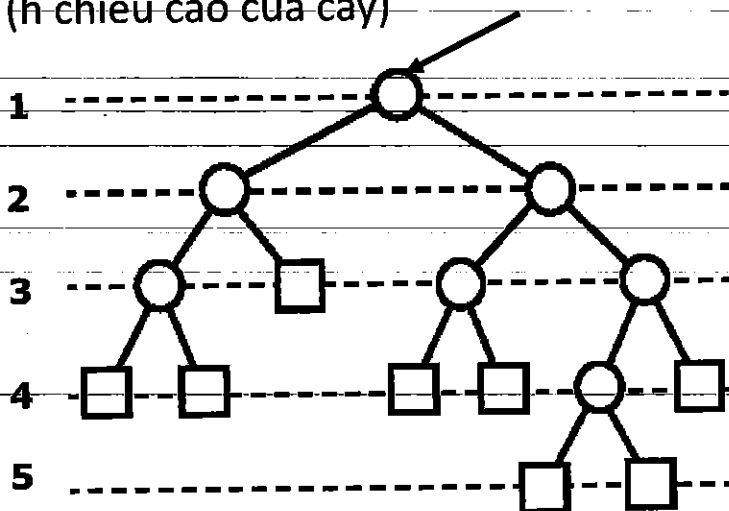
CHƯƠNG 6

CÂY

Tính chất của cây nhị phân

- Số nút tối đa mức i trong cây 2^{i-1}
- Số nút tối đa trong cây là $2^h - 1$ (h chiều cao của cây)

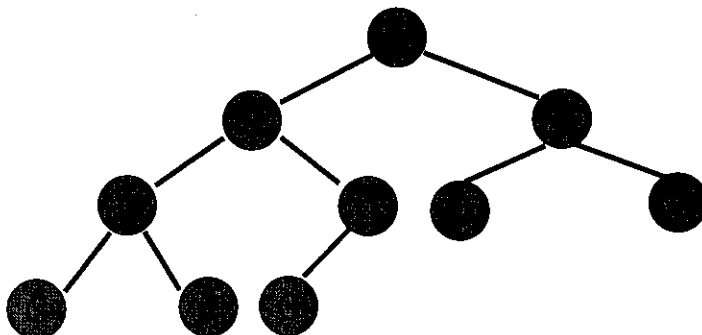
□ Chiều cao của cây $h \geq \log_2 N$ (N là số nút trong cây).



CHƯƠNG 6

CÂY

Cây nhị phân hoàn chỉnh



Các nút ứng với các mức trừ mức cuối đều đạt tối đa, ở mức cuối, các nút đều đạt về phía trái

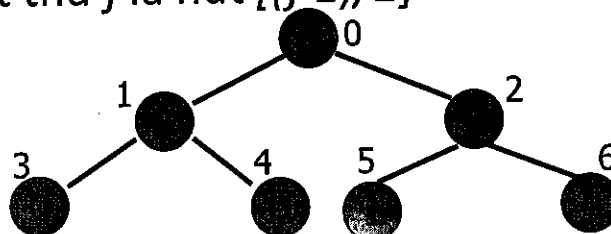


**CHƯƠNG 6****CÂY****Tổ chức lưu trữ cây nhị phân**

- Sử dụng mảng một chiều (lưu trữ kế tiếp)
 - Đánh số thứ tự từ gốc, tại mỗi mức, đánh số các nút từ trái sang phải, từ mức thấp đến mức cao
- Sử dụng liên kết (Lưu trữ liên kết)
 - Quản lý cây thông qua nút gốc (root)
 - Mỗi nút cấp phát động, bao gồm dữ liệu và hai liên kết pLeft, pRight, liên kết tới cây con trái và cây con phải
 - Nút lá có hai liên kết trái phải đều rỗng

**CHƯƠNG 6****CÂY****Lưu trữ kế tiếp cây nhị phân**

- Con của nút thứ i là nút thứ $2i+1$ và $2i+2$
- Cha của nút thứ j là nút $[(j-1)/2]$



V[0] V[1] V[2] V[3] V[4] V[5] V[6]

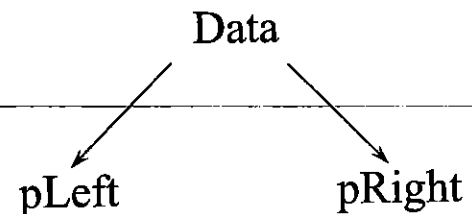


**CHƯƠNG 6****CÂY****Cấu trúc của nút**

```

Class Node {
    int Data;
    Node pLeft; // liên kết đến nút con trái
    Node pRight; // liên kết đến nút con phải
};
Node root = NULL; // gốc của cây

```

**CHƯƠNG 6****CÂY****Phép duyệt cây nhị phân**

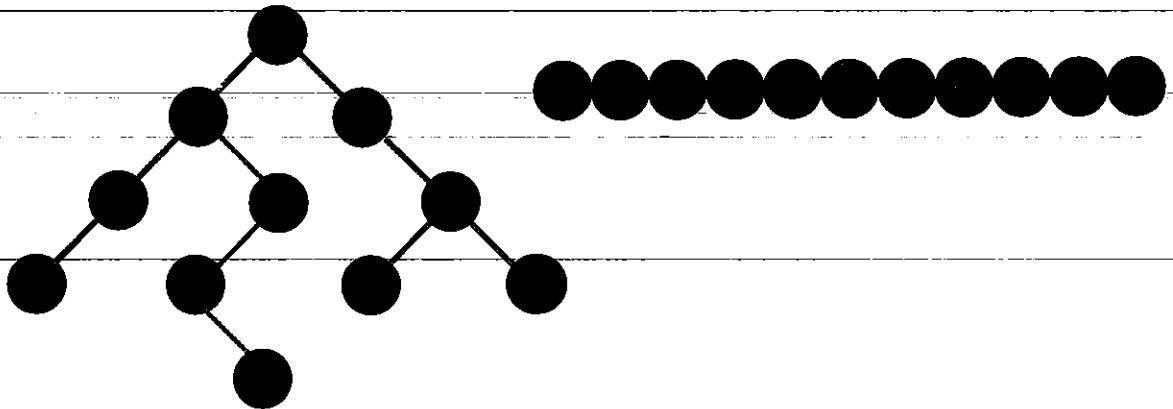
• Định nghĩa

- là phép xử lý các nút trên cây, mỗi nút một lần
- Duyệt cây theo thứ tự trước (preorder)
- Duyệt cây theo thứ tự giữa (inorder)
- Duyệt cây theo thứ tự sau (postorder)



**CHƯƠNG 6****CÂY****Duyệt cây theo thứ tự giữa**

- Duyệt cây theo thứ tự giữa (LNR)
 - Duyệt cây con trái theo thứ tự giữa
 - Thăm gốc
 - Duyệt cây con phải theo thứ tự giữa

**CHƯƠNG 6****CÂY****Duyệt cây theo thứ tự giữa**

```
void inorder(Node root)
```

```
{
```

```
    if (root != NULL) {
```

```
        inorder(root.pLeft);
```

```
        In ra: root.data;
```

```
        indorder(root.pRight);
```

```
    }
```

```
}
```



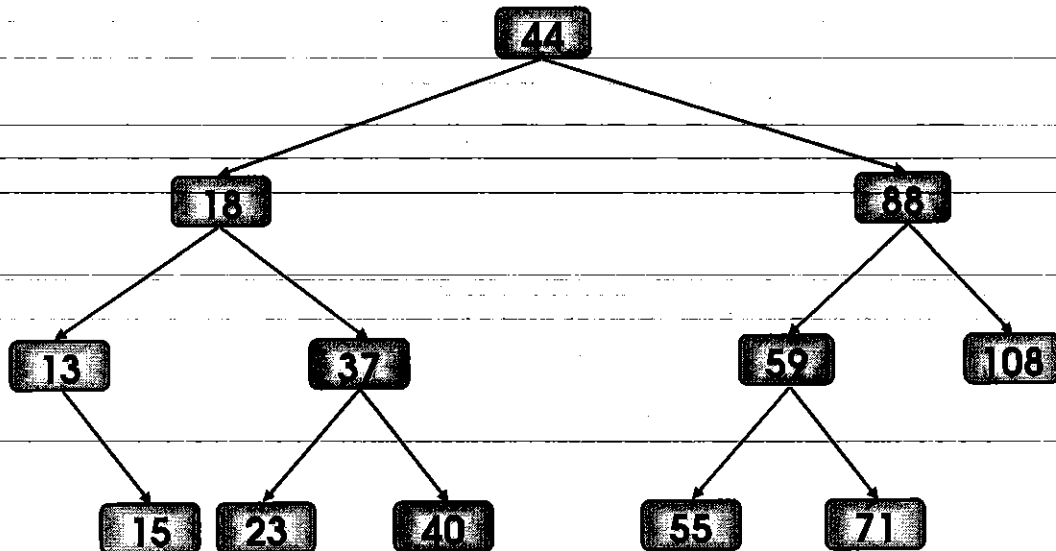


CHƯƠNG 6

CÂY

Cây nhị phân tìm kiếm

- Định nghĩa: (Binary Search Tree – BST)



CHƯƠNG 6

CÂY

Cây nhị phân tìm kiếm

- Khai báo cây

```
Class BSTNode {
```

```
int Data;
```

```
    BSTNode pLeft; //con trỏ đến nút con trái
```

```
    BSTNode pRight; //con trỏ đến nút con phải
```

```
};
```

```
BSTNode root = NULL; //gốc của cây
```





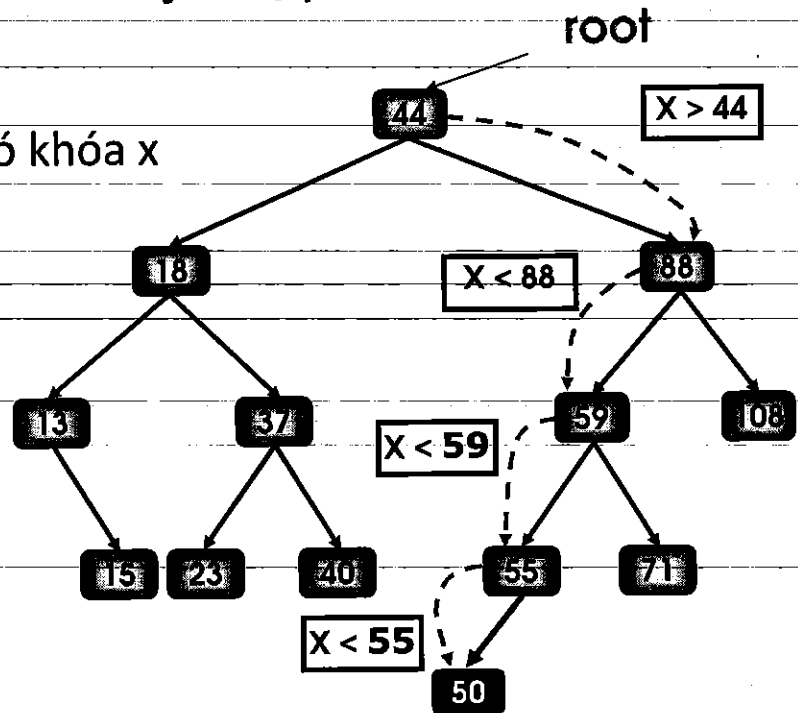
CHƯƠNG 6

CÂY

Thêm một phần tử vào cây nhị phân tìm kiếm

- Thêm vào phần tử có khóa x

Thêm X= 50



CHƯƠNG 6

CÂY

Thêm một phần tử vào cây nhị phân tìm kiếm

```

int Insert(int X, BST_Node root)
{ if (root == NULL)
  { root = new BSTreeNode ;
    if ( root == NULL )
      return -1; // Không thể cấp phát bộ nhớ
    else
      {
        root.Data = X;
        root.pLeft = root.pRight = NULL;
        return 1; // Thêm vào thành công
      }
  }
}
  
```

**CHƯƠNG 6****CÂY****Tìm một nút có khóa X**

```
BSTNode Find( int X, BSTNode root)
{
    if( root == NULL )
        return NULL;
    if ( X < root.data)
        return Find( X, root.pLeft );
    else if ( X > root.data)
        return Find( X, root.pRight );
    else
        return root;
}
```

**CHƯƠNG 6****CÂY****Tìm một nút có khóa X**

- Tìm nút có khóa X, không dùng đệ qui

```
BTSNode Find2(int X, BTSNode root)
{
    BTSNode p = root;
    while (p != NULL)
    {
        if(X == p.data) return p;
        else
            if(x < p.data) p = p.pLeft;
            else
                p = p.pRight;
    }
    return NULL;
}
```





CHƯƠNG 6

CÂY

Xóa một nút có khóa X trên cây BST

• Xóa một nút có khóa X trên cây BST, có ba trường hợp:

- Nút có khóa X là nút lá.
- Nút có khóa X chỉ có 1 con (trái hoặc phải).
- Nút có khóa X có đủ cả 2 con

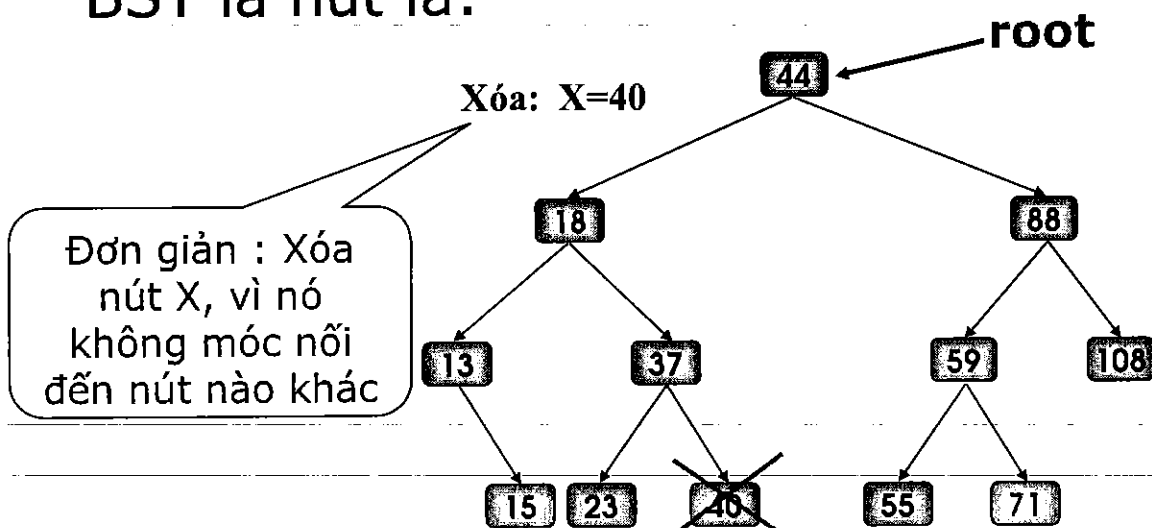


CHƯƠNG 6

CÂY

Xóa một nút có khóa X

Trường hợp 1 : Nút có khóa X trên cây BST là nút lá:





CHƯƠNG 6

CÂY

Cây nhị phân tìm kiếm (Binary Search Tree - BST)

```

int Delete( int X, BSTNode root)
{ BSTNode p;
  if ( root == NULL )
    return 0 ; // cây rỗng, không tìm thấy
  else
    if ( X < root.Data ) // xóa trên cây con trái
      return Delete( X, root.pLeft );
    else
      if ( X > root.Data ) // xóa trên cây con phải
        return Delete( X, root.pRight );
      else // tìm ra nút cần xóa

```



CHƯƠNG 6

CÂY

Cây nhị phân tìm kiếm (Binary Search Tree - BST)

```

if ( root.pLeft && root.pRight ) // Có hai con
{
  p = FindMax(root.pLeft); // tìm nút có khóa lớn nhất trên con trái
  root.Data = p.Data;
  return Delete(root.Data, root.pLeft);
}
else // có một con hoặc không có con
{
  p = root;
  if ( root.pLeft == NULL ) // xử lý như không có con
    root = root.pRight;
  else if ( root.pRight == NULL )
    root = root.pLeft;
  p = null;
}
return 1;
}

```

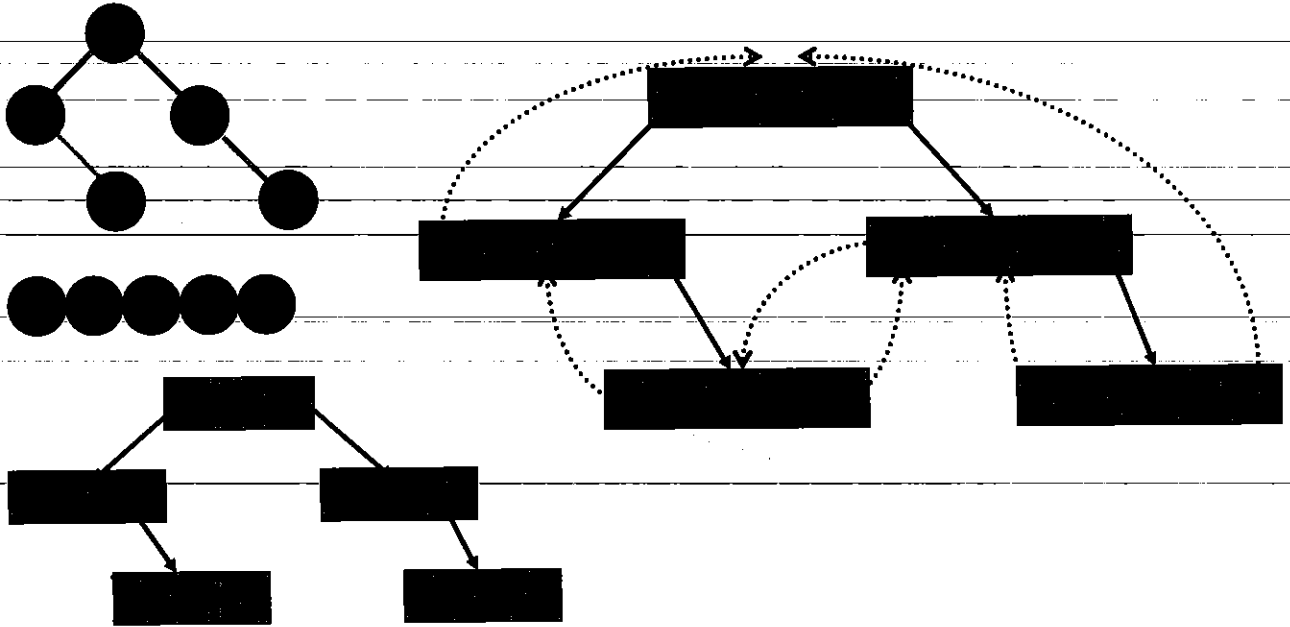




CHƯƠNG 6

CÂY

Cây nhị phân liên kết vòng (NLR)



CHƯƠNG 6

CÂY

Cây tổng quát

- Định nghĩa
 - Cây m phân là cây mà mỗi nút có tối đa m nút con (cây con)
 - Biểu diễn cây m phân bằng liên kết động
 - Mỗi nút có $m+1$ trường, với m mỗi nối
 - Với cây m phân đầy đủ, có $n(m-1)+1$ mỗi liên kết NULL



**CHƯƠNG 6****CÂY**

Phép duyệt cây tổng quát NLR(T)

- Nếu T rỗng, dừng
- Ngược lại, T_1, \dots, T_n là cây con gốc T
 - Thăm gốc của T
 - NLR(T_1), T_1 cây con thứ nhất của gốc T
 - Duyệt cây con T_2, \dots, T_n của T theo thứ tự trước

**CHƯƠNG 6****CÂY**

Phép duyệt cây tổng quát LNR(T)

- Nếu T rỗng, dừng
- Ngược lại, T_1, \dots, T_n là cây con gốc T
 - LNR(T_1), T_1 cây con thứ nhất của gốc T
 - Thăm gốc của T
 - Duyệt cây con T_2, \dots, T_n của T theo thứ tự giữa





Chương 7

ĐỒ THỊ



CHƯƠNG 7

ĐỒ THỊ

Mục tiêu của chương

- Trình bày những kiến thức căn bản về lý thuyết đồ thị, cách biểu diễn, một số thuật toán trên đồ thị
- Một số ứng dụng của đồ thị





CHƯƠNG 7

ĐỒ THỊ

Các khái niệm

- Chu trình (cycle) = $(u_1, u_2, \dots, u_n), u_1 \equiv u_n$
- Đồ thị định hướng (directed graph)
 - $\forall (x, y) \in E : (x, y) \neq (y, x)$
- Đồ thị vô hướng
 - $\forall (x, y) \in E : (y, x) \in E$
 - $(x, y) \equiv (y, x)$

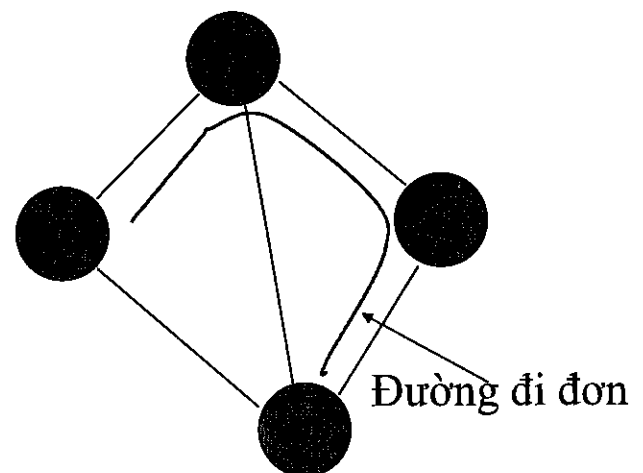
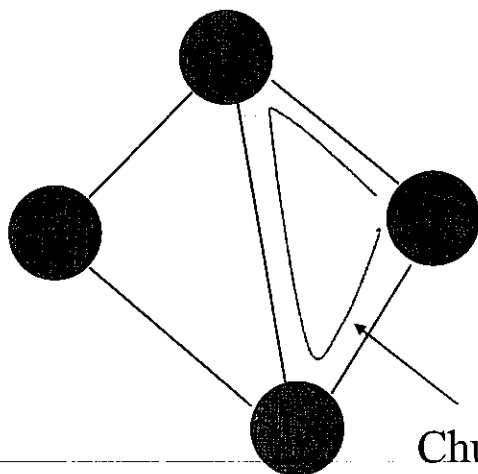


CHƯƠNG 7

ĐỒ THỊ

Các khái niệm

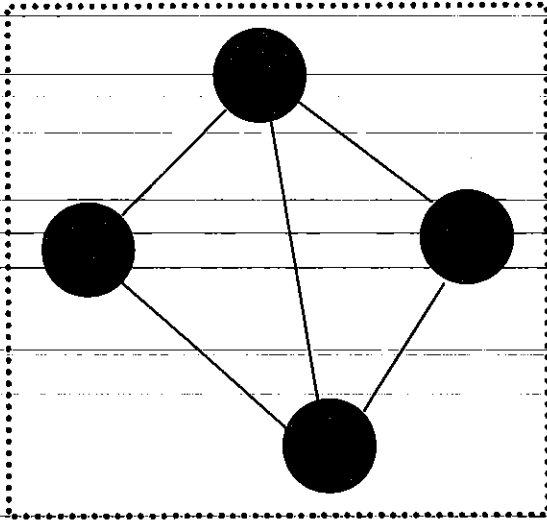
- CBDC là một chu trình



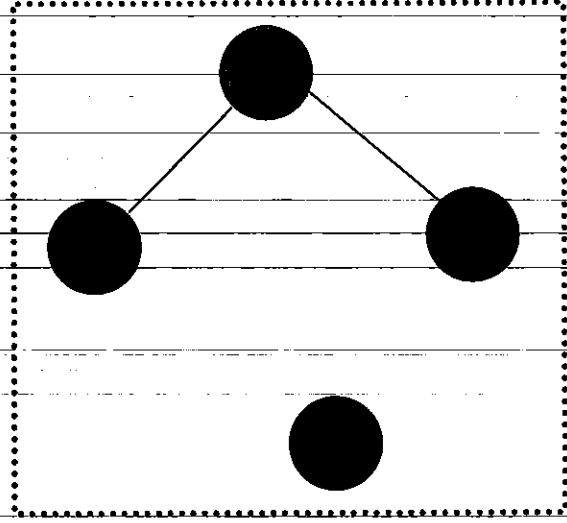


CHƯƠNG 7

ĐỒ THỊ



Đồ thị liên thông



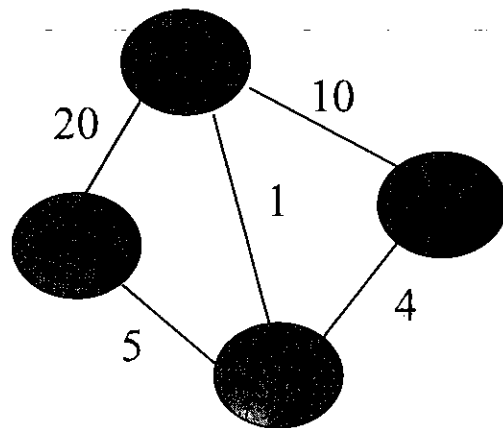
Đồ thị không liên thông



CHƯƠNG 7

ĐỒ THỊ

- Đồ thị có trọng số

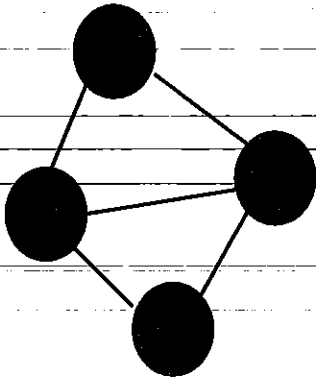




CHƯƠNG 7

ĐỒ THỊ

Biểu diễn bằng ma trận kề



A[i][j]	0	1	2	3
0	0	1	1	0
1	1	0	1	1
2	1	1	0	1
3	0	1	1	0

A =

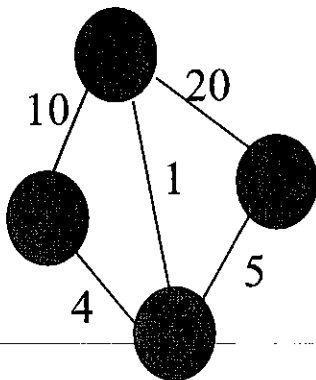
$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$



CHƯƠNG 7

ĐỒ THỊ

Biểu diễn bằng ma trận kề



A[i][j]	0	1	2	3
0	0	20	10	1
1	20	0	0	5
2	10	0	0	4
3	1	5	4	0

A =

$$\begin{pmatrix} 0 & 20 & 10 & 1 \\ 20 & 0 & 0 & 5 \\ 10 & 0 & 0 & 4 \\ 1 & 5 & 4 & 0 \end{pmatrix}$$

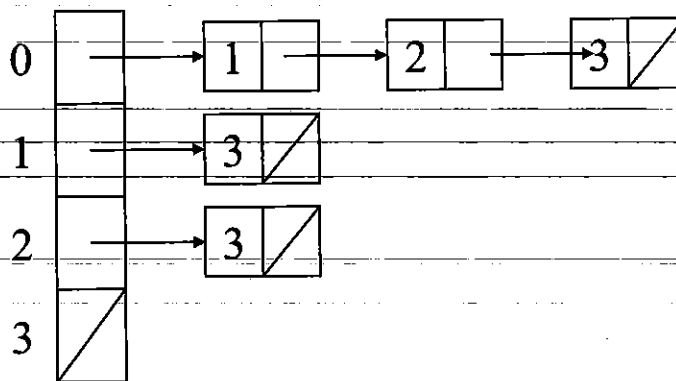
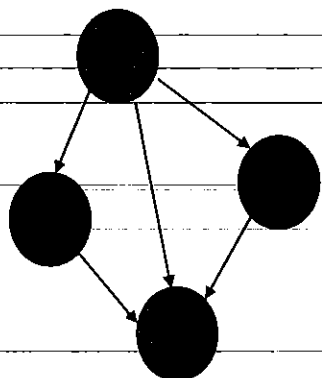




CHƯƠNG 7

ĐỒ THỊ

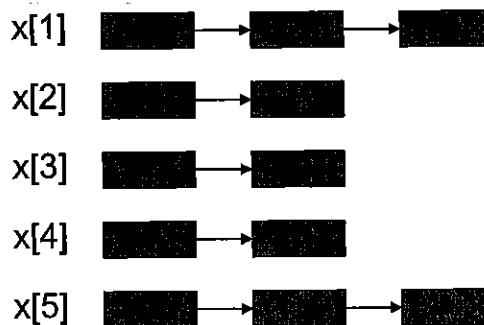
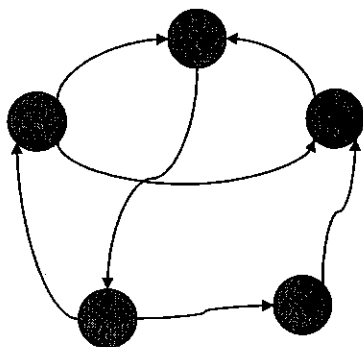
Biểu diễn đồ thị bằng danh sách kề



CHƯƠNG 7

ĐỒ THỊ

Biểu diễn đồ thị bằng danh sách kề





CHƯƠNG 7

ĐỒ THỊ

Phép duyệt đồ thị

Từ một đỉnh, liệt kê tất cả các đỉnh của đồ thị

- Phép tìm kiếm theo chiều sâu
 - Depth first search
- Phép tìm kiếm theo chiều rộng
 - Breadth first search



CHƯƠNG 7

ĐỒ THỊ

Duyệt theo chiều sâu

- Xuất phát từ đỉnh V (coi như được thăm). Xét 1 đỉnh w là lân cận của V nhưng chưa được thăm.
- Nếu 1 đỉnh V đã được thăm mà mọi đỉnh lân cận của nó đều được thăm thì ta sẽ quay ngược lên đỉnh cuối cùng vừa được thăm mà nó có đỉnh w lân cận chưa được thăm và lặp lại cho tới khi không một nút nào không được thăm.



**CHƯƠNG 7****ĐỒ THỊ**

Duyệt theo chiều rộng

Chọn 1 đỉnh xuất phát, giả sử là V , ta lần lượt thăm tất cả các lân cận đỉnh của V (nhưng chưa được thăm), sau đó lại lần lượt thăm tất cả các lân cận của các lân cận của V mà cũng chưa được thăm.

**CHƯƠNG 7****ĐỒ THỊ**

Giải thuật :

Procedure BFS(v) // v : đỉnh xuất phát VISITED(i)=1
=> i được thăm ...

VISITED(v)=1

Khởi tạo q với v đã được nạp vào ;

While (Q không rỗng)

{call CQDELETE(v , Q);

//lấy đỉnh v ra khỏi Q



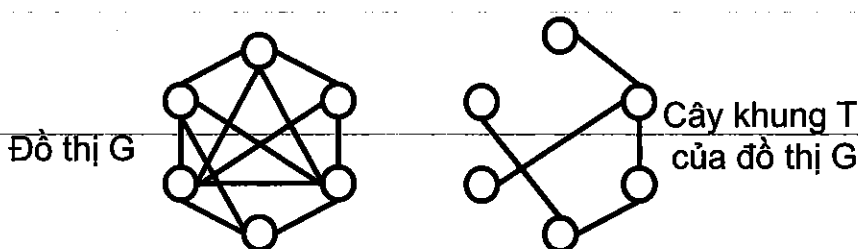


CHƯƠNG 7

ĐỒ THỊ

Cây khung (Spanning tree)

- T bao gồm các cung thuộc một phép duyệt từ một đỉnh đến các đỉnh còn lại trong V
- Giá của cây khung $T =$ tổng trọng số của các cung thuộc E



CHƯƠNG 7

ĐỒ THỊ

Cây khung (Spanning tree)

Chú ý

- Một đồ thị G có thể có nhiều cây khung
- Cây khung theo chiều rộng, theo chiều sâu
- Các cung trong cây khung không tạo nên chu trình
 - Giữa hai đỉnh trong một cây khung chỉ tồn tại duy nhất một đường đi từ đỉnh này đến đỉnh kia
- Nếu đồ thị có n đỉnh, thì cây khung có $n-1$ cạnh





CHƯƠNG 7

ĐỒ THỊ

Thuật toán Kruskal

Để kiểm tra xem có tạo ra chu trình trong T hay không, chúng ta xem hai đỉnh của cung được thêm có thuộc tập các đỉnh hiện có trong T không, nếu có, nghĩa là sẽ tạo nên chu trình



CHƯƠNG 7

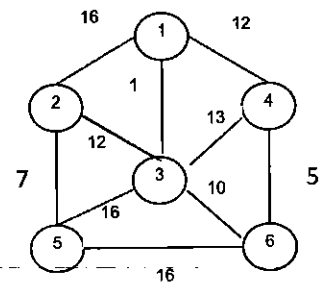
ĐỒ THỊ

Thuật toán Kruskal

Các cung được xét để đưa vào T theo thứ tự :

(1, 3), (4,6), (2,5), (3,6), (2,3), (3,4), (1,4), (1,2), (3,5), (5,6)
 1 5 7 10 12 12 13 16 16 16

4 cung đầu được nhận vào T, cung (3,4), (1,4) bị loại vì tạo chu trình trong T, cung (2,3) được tiếp nhận, cung (1,2), (3,5), (5,6) bị loại vì tạo chu trình trong T.
 => Còn (1, 3), (4,6), (2,5), (3,6), (2,3)
 Cây có 5 cành -> giá trị cực tiểu là 35



**CHƯƠNG 7****ĐỒ THỊ****Thuật toán Dijkstra**

- Xét đồ thị có hướng $G=(V,E)$, với $|V|=n$
- Ma trận trọng số $d[u,v] \geq 0, \forall (u,v) \in E$
- $s \in V$ là điểm xuất phát
- $H[v]$ = chiều dài cực tiểu từ s đến v
($v \in V$)

**CHƯƠNG 7****ĐỒ THỊ****Thuật toán Dijkstra****Bắt đầu duyệt từ đỉnh s** **Gán giá trị cho $H[v]$**

- $H[v] = d(s,v)$, nếu $(s,v) \in E$
- $H[v] = \infty$, nếu ngược lại

Lặp lại cho đến khi duyệt hết các đỉnh

- Chọn đỉnh w chưa duyệt có $H[w]$ nhỏ nhất
- Duyệt đỉnh w này
- Với các đỉnh t chưa duyệt khác
 - $H[t] = \min(H[t], H[w] + d(w,t))$





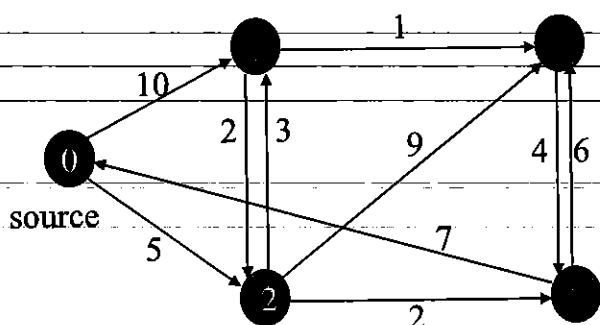
CHƯƠNG 7

ĐỒ THỊ

Thuật toán Dijkstra

step 1: tìm đường đi ngắn nhất từ 0

- node 2 được chọn



node	from node V_0 to other nodes			
V_1	10			
V_2	5			
V_3	∞			
V_4	∞			
best	V_2			



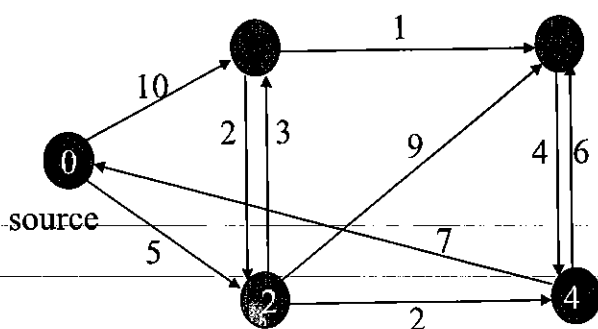
CHƯƠNG 7

ĐỒ THỊ

Thuật toán Dijkstra

step 2: Tính toán lại các đường đi đến tất cả các đỉnh

- Tìm đường đi ngắn nhất đến node 0. Node 4 được chọn



node	from node V_0 to other nodes			
V_1	10	8		
V_2	5	5		
V_3	∞	14		
V_4	∞	7		
best	V_2	V_4		



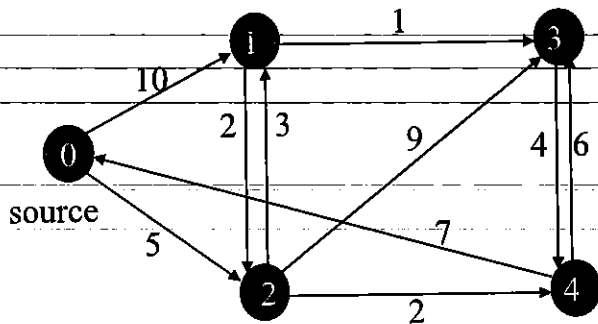


CHƯƠNG 7

ĐỒ THỊ

Thuật toán Dijkstra

- Chúng ta có tất cả các đường đi từ v_0



node	from node V_0 to other nodes			
V_1	10	8 (0,2)	8 (0,2)	8
V_2	5 (0,2)	5	5	5
V_3	∞	14 (0,2,3)	13 (0,2,4,3)	9 (0,2,1,3)
V_4	∞	7 (0,2,4)	7	7
best	V_2	V_4	V_1	V_3



Chương 8

SẮP XẾP



CHƯƠNG 8 SẮP XẾP

Ví dụ :

Sắp xếp một danh sách sinh viên theo vần A, B, C

Định nghĩa

Sắp xếp là quá trình tổ chức lại tập dữ liệu theo một trật tự tăng dần hay giảm dần



CHƯƠNG 8: SẮP XẾP

Các phương pháp sắp xếp

Các thuật toán cơ bản

- Thuật toán “Selection sort”
- Thuật toán “Insertion sort”
- Thuật toán “Bubble sort”
- Thuật toán “Heap sort”
- Thuật toán “Quick sort”





CHƯƠNG 8: SẮP XẾP

Sắp xếp lựa chọn (selection sort)

```

void selectionSort(int a[], int n)
{
    int i, j, min, minidx, temp;
    for (i = 0; i < (n - 1); i++)
    {
        min = a[i];
        minidx = i;
        for (j = i + 1; j < n; j++)
        {
            if (a[j] < min)
            {
                min = a[j];
                minidx = j;
            }
        }
        if (min < a[i])
        {
            temp = a[i];
            a[i] = min;
            a[minidx] = temp;
        }
    }
}

```



CHƯƠNG 8: SẮP XẾP

Độ phức tạp tính toán

Ở bước thứ i , có $(n-i)$ lần so sánh, với $i=1 \dots n-1$

$$(n-1) + (n-2) + \dots + 1 = n(n-1)/2 = O(n^2)$$

Thời gian thực hiện giải thuật

$$T(n) \sim O(n^2)$$





CHƯƠNG 8: SẮP XẾP

Sắp xếp chèn (Insert sort)

Giải thuật

```

void InsertionSorting(int a[], int n){
    int x,i,j;
    for (i=1;i<n;i++){
        x=a[i];
        j=i-1;
        while (x<a[j]) && (j>=0){
            a[j+1]=a[j];
            j=j-1;
        }
        a[j+1]=x;
    }
}

```



CHƯƠNG 8: SẮP XẾP

Sắp xếp chèn (Insert sort)

- Độ phức tạp tính toán
 - Ở bước thứ i , có tối đa $i-1$, tối thiểu 1 phép so sánh
 - Thời gian thực hiện giải thuật $T(n) \sim O(n^2)$
 - Trường hợp xấu nhất có:
 - $1 + 2 + 3 + \dots + (n-1) = n(n-1)/2 = O(n^2)$ phép so sánh và dịch chuyển
 - Trường hợp tốt nhất (mảng đã có thứ tự tăng dần): $O(n)$ phép so sánh và 0 phép dịch chuyển





CHƯƠNG 8: SẮP XẾP

Sắp xếp nổi bọt (Bubble Sort)

```
void BubbleSorting(int a[], int n) {  
    int tmp;  
    for (int i=0; i<n; i++) {  
        for (int j=0; j<n-i-1; j++) {  
            if (a[j]>a[j+1]) {  
                tmp=a[j+1];  
                a[j+1]=a[j];  
                a[j]=tmp;  
            }  
        }  
    }  
}
```



CHƯƠNG 8: SẮP XẾP

Sắp xếp nổi bọt (Bubble Sort)

Độ phức tạp tính toán

- Ở bước thứ i , có $n-i$ phép so sánh
- Thời gian thực hiện giải thuật
 $T(n) \sim O(n^2)$

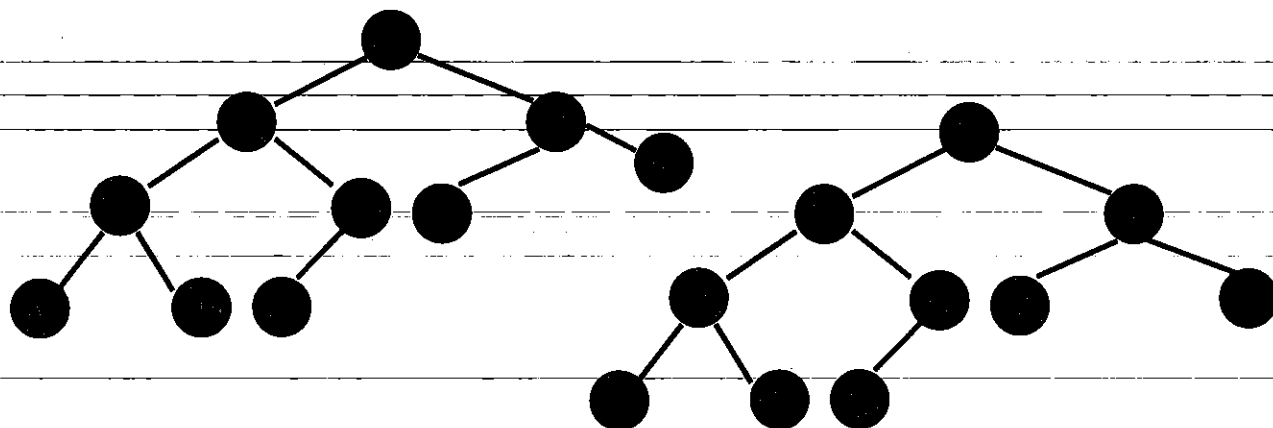




CHƯƠNG 8 SẮP XẾP

Heap sort

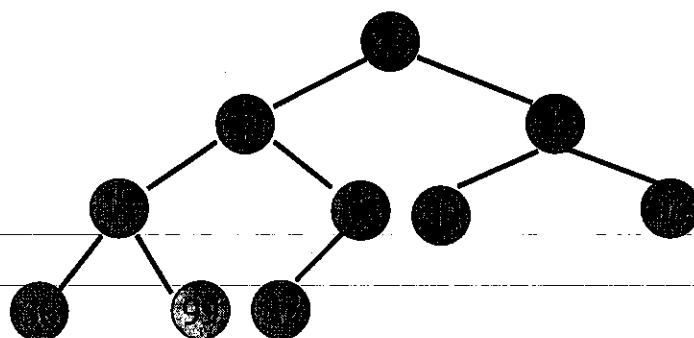
Xét dãy 43,23,71,11,65,58,94,36,99,87



CHƯƠNG 8 SẮP XẾP

Heap sort

- Phương pháp tạo HEAP từ đáy lên
- Việc tạo HEAP bắt đầu từ các nút trong (nút số 0 đến nút số $[n/2]-1$)





CHƯƠNG 8 SẮP XẾP

Heap sort

```
void Heapsort()
```

```
{  
    int tmp;  
    makeheap(a,n)  
    for (int i=n-1;i>0;i--){  
        tmp=a[0];a[0]=a[i];a[i]=tmp;  
        setupHeap(a,0,i);  
    }  
}
```



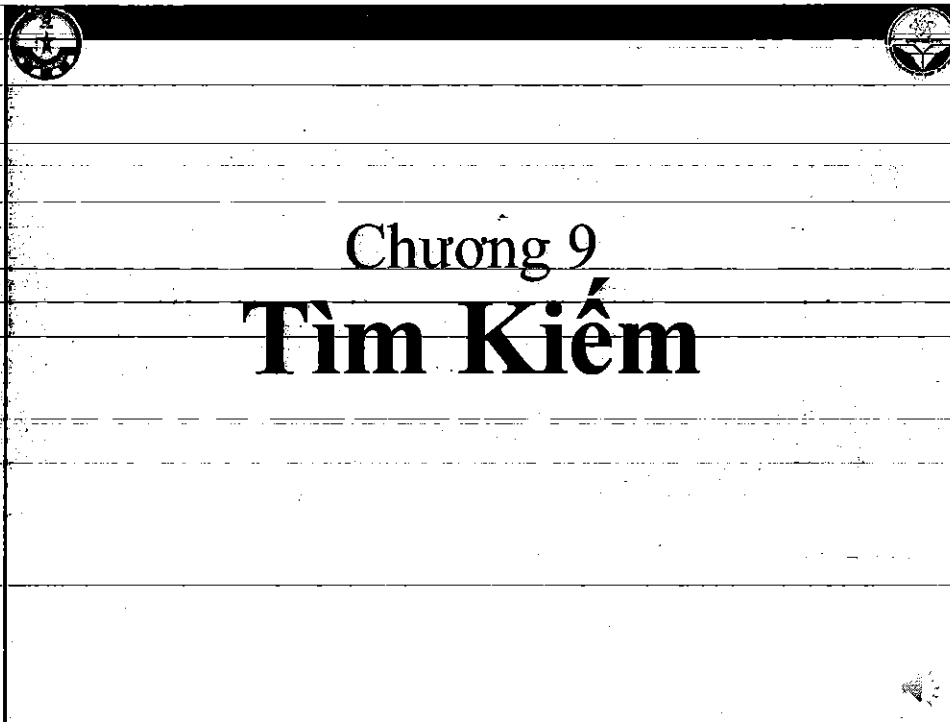
CHƯƠNG 8 SẮP XẾP

Sắp xếp nhanh (Quick sort)

Ý tưởng

- Xét một dãy n phần tử a_1, a_2, \dots, a_n
- (1) chọn phần tử $x = a[(n+1) \div 2]$ làm khóa
- (2) đi từ hai đầu của dãy, nếu gặp một cặp $a[i] \geq x \geq a[j]$ thì hoán vị hai phần tử này
- (3) tăng $i = i + 1$, giảm $j = j - 1$
- (4) lặp lại (2) cho đến khi $i > j$ (kết quả thu được phân đoạn AxB)
- (5) lặp lại (1)-(4) với hai phân đoạn A và B
- Kết thúc khi tất cả các phân đoạn thu được có chiều dài là 1





CHƯƠNG 9
Tìm Kiếm

Mục tiêu

- Trình bày các thuật toán thông dụng cho việc tìm kiếm (**Tìm tuần tự, tìm nhị phân**)
- Minh họa các thuật toán
- Đánh giá thuật toán



CHƯƠNG 9



Tìm Kiếm

Tìm kiếm tuần tự (sequential searching)

Ý tưởng

- Lần lượt tìm kiếm từ bản ghi đầu tiên cho đến khi tìm thấy, hoặc không còn phần tử để tìm kiếm
- Thực hiện tìm kiếm trên mảng/danh sách liên kết đơn



CHƯƠNG 9



Tìm Kiếm

Tìm kiếm tuần tự (sequential searching)

- Giải thuật

```
bool SequentialSearch(int a[],int n,int x){
    for (int i=0;i<n;i++){
        if (a[i]==x) return true;
    }
    return false;
}
```



CHƯƠNG 9

Tìm Kiếm



Tìm kiếm nhị phân

```
int BinarySearch(int a[],int l, int r,int x){
    int m
    while (l<=r){
        int m=(l+r)/2;
        if (a[m]==x) return m;
        if (x<a[m]) r=m-1;
        if (x>a[m]) l=m+1;
    }
    return -1;
}
```



CHƯƠNG 9

Tìm Kiếm



Tìm kiếm nhị phân

Giải thuật

```
int BinarySearch1(int a[],int l,int r,int x)
{
    int m=(l+r)/2;
    if (a[m]==x) return m;
    if ( x <a[m]) return BinarySearch1(a,l,m-1,x);
    if (x > a[m]) return BinarySearch1(a,m+1,r,x);
    return -1;
}
```



DỰ TOÁN KINH PHÍ ĐỀ TÀI KH&CN CẤP TRƯỜNG NĂM 2022

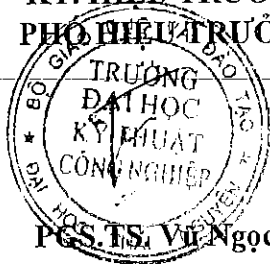
Tên đề tài: Xây dựng video bài giảng môn Cấu trúc dữ liệu và giải thuật

Chủ nhiệm đề tài: Nguyễn Thị Hương

ĐVT: VND

STT	Nội dung	Dự toán			
		Người thực hiện	Số ngày công	Hệ số tiền công theo ngày (2)*	Thành tiền
1	Mục chi tiền công lao động tham gia trực tiếp (1)				
1.1	Xây dựng thuyết minh đề tài	Nguyễn Thị Hương	0.5	0.45	335.250
1.2	Xây dựng các file bài giảng trình chiếu Powerpoint cho môn học	Nguyễn Thị Hương	1	0.45	670.500
1.3	Ghi hình các bài giảng chương 1,2	Nguyễn Thị Hương	1	0.45	670.500
1.4	Ghi hình các bài giảng chương 3,4	Nguyễn Thị Hương	1	0.45	670.500
1.5	Ghi hình các bài giảng chương 5,6	Nguyễn Thị Hương	1	0.45	670.500
1.6	Ghi hình các bài giảng chương 7	Nguyễn Thị Hương	1	0.45	670.500
1.7	Ghi hình các bài giảng chương 8,9	Nguyễn Thị Hương	1	0.45	670.500
1.8	Hậu kỳ chỉnh sửa các video	Nguyễn Thị Hương	1	0.45	670.500
1.9	Viết báo cáo tổng kết	Nguyễn Thị Hương	0.5	0.45	335.250
	Tổng 1		8		5.364.000
2	Mục chi khác				
	Phô tô, in ấn				36.000
	Tổng 2				36.000
	Tổng (1+2)				5.400.000

Cơ quan chủ trì
KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG



PGS. TS. Vũ Ngọc Pi

TRƯỜNG PHÒNG KH&CN&HTQT

PGS. TS. Phạm Thành Long

CHỦ NHIỆM ĐỀ TÀI

Nguyễn Thị Hương

TRƯỜNG PHÒNG KH-TC

