

Mục lục:

A. Nội dung chính:	3
B. Nội dung chi tiết:	3
1. Giới thiệu về danh sách liên kết:	3
2. Danh sách liên kết đơn:	3
☑ Đặc điểm tổ chức:	3
☑ Cấu trúc lưu trữ:	4
☑ Phép toán:	4
✎ 1. Phép bổ sung vào sau nút M:	4
✎ 2. Phép loại bỏ 1 nút khỏi danh sách:	5
✎ 3. Phép duyệt danh sách:	7
✎ 4. Phép bổ sung vào trước nút M:	7
3. Danh sách liên kết kép:	9
☑ Đặc điểm tổ chức:	9
☑ Cấu trúc lưu trữ:	10
☑ Phép toán:	10
✎ 1. Phép bổ sung vào trước:	10
✎ 2. Phép loại bỏ 1 nút ra khỏi danh sách liên kết kép:	12
✎ 3. Duyệt danh sách liên kết kép:	13
✎ 4. Bổ sung phần tử dữ liệu x vào sau nút M:	14
4. Cài đặt Ngăn xếp bằng CTLT phân tán (DSLKD).	15
☑ Đặc điểm tổ chức:	16
☑ Cấu trúc lưu trữ:	16
☑ Phép toán:	16
✎ 1. Bổ sung vào ngăn xếp:	16
✎ 2. Loại bỏ nút đỉnh khỏi ngăn xếp:	17
✎ 3. Kiểm tra ngăn xếp rỗng:	18

5. Cài đặt Hàng đợi bằng CTLT phân tán.	18
<input checked="" type="checkbox"/> Đặc điểm tổ chức:	18
<input checked="" type="checkbox"/> Cấu trúc lưu trữ:	19
<input checked="" type="checkbox"/> Phép toán:.....	19
1. Bổ sung vào hàng đợi:.....	19
2. Loại bỏ phần tử ra khỏi hàng đợi:	20
3. Kiểm tra hàng đợi rỗng:	21
Câu hỏi ôn tập chương 3:	22

ĐỀ CƯƠNG CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

CHƯƠNG 3: DANH SÁCH LIÊN KẾT

A. Nội dung chính:

1. Giới thiệu về danh sách liên kết.
2. Danh sách liên kết đơn.
3. Danh sách liên kết kép.
4. Cài đặt Ngăn xếp bằng CTLT phân tán (DSLKD).
5. Cài đặt Hàng đợi bằng CTLT phân tán (DSLKD).

B. Nội dung chi tiết:

1. Giới thiệu về danh sách liên kết:

- Danh sách là gì?
- Danh sách tuyến tính?
- Trong CTLT này các phần tử dl của danh sách được lưu trữ trong các ô nhớ nằm rải rác khắp nơi trong bộ nhớ nhưng có liên kết với nhau về mặt địa chỉ.
- Một ô nhớ được gọi là một Node (nút)
- Giữa các nút chỉ có 1 liên kết -> danh sách liên kết đơn
2 -> danh sách liên kết kép
- Để cài đặt được ct này -> cơ chế tạo và hủy nút phải thực hiện cấp phát bộ nhớ động

2. Danh sách liên kết đơn:

☒ Đặc điểm tổ chức:

- Danh sách là gì?
- Danh sách tuyến tính là gì?
- Danh sách liên kết là gì?

➔ Danh sách liên kết là danh sách tuyến tính khi sử dụng CTLT phân tán.

Trong CTLT này thì các phần tử dữ liệu được lưu trữ trong các nút nhớ.

➔ Nếu giữa các nút nhớ có 1 liên kết thì ta có danh sách liên kết đơn.

- Trong DSLKĐ, mỗi nút nhớ có cấu trúc gồm 2 trường:

- Trường Infor: chứa thông tin phần tử dữ liệu.
- Trường Link: chứa địa chỉ của nút đứng sau.
- Vẽ hình:

Infor	Link
-------	------

- Nút cuối cùng không có nút đứng sau nên trường Link = \emptyset .
- Để truy nhập vào tất cả các nút trong danh sách, dùng con trỏ F trỏ đến nút đầu tiên trong danh sách.
- Khi danh sách rỗng thì F = \emptyset .

Vẽ hình!!!

☑ Cấu trúc lưu trữ:

- Trong DSLKĐ, mỗi nút nhớ có cấu trúc gồm 2 trường:

- Trường Infor: chứa thông tin phần tử dữ liệu.
- Trường Link: chứa địa chỉ của nút đứng sau.
- Vẽ hình:

Infor	Link
-------	------

- Nút cuối cùng không có nút đứng sau nên trường Link = \emptyset .
- Để truy nhập vào tất cả các nút trong danh sách, dùng con trỏ F trỏ đến nút đầu tiên trong danh sách.
- Khi danh sách rỗng thì F = \emptyset .

☑ Phép toán:

1. Phép bổ sung vào sau nút M:

- Vào: F, M, x.
- Ra: Không có.

{ Thủ tục này bổ sung phần tử dữ liệu x vào sau nút M trong DSLKĐ F }

Procedure Insert(Var F; M, x)

1. Tạo nút mới

$N \leftarrow \text{AVAIL};$

$\text{Infor}(N) := x;$

$\text{Link}(N) := \emptyset;$

2. Thực hiện bổ sung

If $F = \emptyset$ then $F := N;$

Else

Begin

$\text{Link}(N) := \text{Link}(M);$

$\text{Link}(M) := N;$

End;

Return

2. Phép loại bỏ 1 nút khỏi danh sách:

- Vào: F, M, x.
- Ra: Không có.

{ Thủ tục này loại bỏ nút trỏ bởi M trong DSLKĐ F }

Procedure Delete(Var F; M)

1. Kiểm tra danh sách rỗng

If $F = \emptyset$ then

begin

Write('Danh sách rỗng!');

Return;

End;

2. Nút trở bởi M là nút đầu tiên trong danh sách

If $M = F$ then

Begin

$F := \text{Link}(F)$;

$M \Rightarrow \text{AVAIL}$;

Return;

End;

3. Tìm đến nút đứng trước M

$P := F$;

While $\text{Link}(P) \neq M$ do

$P := \text{Link}(P)$;

4. Loại bỏ nút trở bởi M và hủy nút M

$\text{Link}(P) := \text{Link}(M)$;

$M \Rightarrow \text{AVAIL};$

Return

3. Phép duyệt danh sách:

- Vào: F.
- Ra: Không có.

{Thủ tục này duyệt DSLKĐ F và đưa ra các phần tử dữ liệu trong danh sách}

Procedure Display(F)

P := F;

While P \neq NULL do

Begin

Write(Infor(P));

P := Link(P);

End;

Return

4. Phép bổ sung vào trước nút M:

- Vào: F, M, x.
- Ra: Không có.

{Thủ tục này bổ sung phần tử dữ liệu x vào trước nút M trong DSLKĐ F}

Procedure PreInsert(Var F; M, x)

1. Tạo nút mới

$N \leq \text{AVAIL};$

Infor(N):=x;

Link(N):= \emptyset ;

2. Trường hợp danh sách rỗng

if $F=\emptyset$ then

Begin

F:=N;

Return;

end

3. Trường hợp M trở đến nút đầu tiên trong danh sách

if $M=F$ then

begin

Link(N):=F;

F:=N;

Return;

end;

4. Trường hợp còn lại

+ Tìm đến nút đứng trước M

P:=F;

while $\text{Link}(P) \neq M$ do P:=Link(P);

+ Nối nút mới vào nút trước M, sau nút P

$$\text{Link}(P) := N;$$

$$\text{Link}(N) := M;$$

Return

3. Danh sách liên kết kép:

☑ Đặc điểm tổ chức:

- Danh sách là gì?
- Danh sách tuyến tính là gì?
- Danh sách liên kết là gì?
- ➔ Danh sách liên kết là danh sách tuyến tính khi sử dụng CTLT phân tán.

Trong CTLT này thì các phần tử dữ liệu được lưu trữ trong các nút nhớ.

➔ Nếu giữa các nút nhớ có 2 liên kết thì ta có danh sách liên kết kép.

- Trong danh sách liên kết kép thì mỗi nút nhớ gồm 3 trường:
 - + Trường Infor: chứa thông tin phần tử dữ liệu.
 - + Trường Left: con trỏ trỏ đến nút đứng trước.
 - + Trường Right: con trỏ trỏ đến nút đứng sau.
 - + Vẽ hình:

Left	Infor	Right
------	-------	-------

- Nút cực phải không có nút đứng sau nên trường Right = \emptyset .
- Nút cực trái không có nút đứng trước nên trường Left = \emptyset .
- Để truy nhập vào danh sách cả 2 chiều thì dùng 2 con trỏ:
 - + Con trỏ L trỏ đến nút cực trái.
 - + Con trỏ R trỏ đến nút cực phải.
- Vẽ hình!!!
- Khi danh sách rỗng thì $L = R = \emptyset$.

☑ Cấu trúc lưu trữ:

- Trong danh sách liên kết kép thì mỗi nút nhớ gồm 3 trường:
 - + Trường Infor: chứa thông tin phần tử dữ liệu.
 - + Trường Left: con trỏ trỏ đến nút đứng trước.
 - + Trường Right: con trỏ trỏ đến nút đứng sau.
 - + Vẽ hình:

Left	Infor	Right
------	-------	-------

- Nút cực phải không có nút đứng sau nên trường Right = \emptyset .
Nút cực trái không có nút đứng trước nên trường Left = \emptyset .
- Để truy nhập vào danh sách cả 2 chiều thì dùng 2 con trỏ:
 - o Con trỏ L trỏ đến nút cực trái.
 - o Con trỏ R trỏ đến nút cực phải.
- Vẽ hình!!!
- Khi danh sách rỗng thì $L = R = \emptyset$.

☑ Phép toán:

1. Phép bổ sung vào trước:

- Vào: L, R, M, x.
- Ra: Không có.

{ Thủ tục này bổ sung phần tử dữ liệu x vào trước nút M trong DSLKK F, R }

Procedure Insert(Var L, R; M, x)

1. Tạo nút mới

$N \leftarrow \text{AVAIL};$

$\text{Infor}(N) := x;$

Left(N) := \emptyset ;

Right(N) := \emptyset ;

2. Trường hợp danh sách rỗng

If $L = R = \emptyset$ then

Begin

L := R := N;

Return;

End;

3. M trở đến nút cực trái

If $M = L$ then

Begin

Right(N) := L;

Left(N) := N;

L := N;

Return;

End;

4. Bổ sung vào giữa danh sách trước M

Right(Left(M)) := N;

Left(N) := Left(M);

Left(M) := N;

Right(N) := M;

Return

2. Phép loại bỏ 1 nút ra khỏi danh sách liên kết kép:

- Vào: L, R, M.
- Ra: Không có.

{Thủ tục này loại bỏ nút trỏ bởi M trong DSLKK L, R}

Procedure Delete(Var L, R; M)

1. {Trường hợp danh sách rỗng!}

If $L=R=\emptyset$ then

Begin

Write('Danh sach rong!');

Return;

End;

2. {Loại bỏ}

Case

$L=R$: {Danh sách chỉ còn 1 nút}

Begin

$L:=\emptyset$;

$R:=\emptyset$;

End;

$M=L: \{Nút\ cực\ trái\ bị\ loại\}$

Begin

$L:=Right(L);$

$Left(L):=\emptyset;$

End;

$M=R: \{Nút\ cực\ phải\ bị\ loại\}$

Begin

$R:=Left(R);$

$Right(R):=\emptyset;$

End;

Else

Begin

$Right(Left(M)):=Right(M);$

$Left(Right(M)):=Left(M);$

End;

End Case

3. {Hủy nút M}

$M \Rightarrow AVAIL;$

Return

3. Duyệt danh sách liên kết kép:

- Vào: L, R.
- Ra: Không có.

{ Thủ tục này duyệt DSLKK L,R và đưa ra các phần tử dữ liệu trong danh sách theo thứ tự từ trái qua phải }

Procedure Display(L, R)

P:=L;

while P != \emptyset do

begin

write(Infor(P));

P:=Right(P);

end;

Return

4. Bổ sung phần tử dữ liệu x vào sau nút M:

- Vào: L, R, M, x.
- Ra: Không có.

{ Thủ tục này bổ sung phần tử x vào sau nút M trong DSLK kép L, R }

Procedure PostInsert(Var L, R; M, x)

1. Tạo nút mới

N<=AVAIL;

Left(N):=Right(N):= \emptyset ;

2. Trường hợp danh sách rỗng

if $L = \emptyset$ then

Begin

$L := R := N;$

Return;

end

3. Trường hợp M trở đến nút cực phải

if $M = R$ then

begin

$Left(N) := R;$

$Right(R) := N;$

$R := N;$

Return;

end;

4. Trường hợp còn lại

$Right(N) := Right(M);$

$Left(Right(M)) := N;$

$Right(M) := N;$

$Left(N) := M;$

Return

4. Cài đặt Ngăn xếp bằng CTLT phân tán (DSLKD).

☑ Đặc điểm tổ chức:

Đã được trình bày ở chương 2! (5 ý cần phải trình bày!)

☑ Cấu trúc lưu trữ:

- Trong CTLT này, các phần tử dữ liệu được lưu trữ trong các nút nhớ nằm rải rác khắp nơi trong bộ nhớ, nhưng có liên kết với nhau về mặt địa chỉ. Mỗi nút nhớ gồm có 2 trường:
 - + Trường Infor: chứa thông tin phần tử dữ liệu.
 - + Trường Link: chứa địa chỉ của các nút dưới.
 - + Vẽ hình ngăn xếp dạng đứng, có các nút nhớ.
- Nút dưới cùng không có nút đứng sau nên trường Link = \emptyset .
- Dùng con trỏ T trỏ đến nút trên cùng của ngăn xếp.
- Khi T thay đổi thì ngăn xếp hoạt động.
 - + Khi bỏ sung thì tạo nút mới chứa phần tử dữ liệu và nối vào trên nút T rồi cho T trỏ đến nút mới.
 - + Khi loại bỏ thì loại bỏ nút T và cho T trỏ đến nút bên dưới.
- Khi ngăn xếp rỗng thì $T = \emptyset$.

☑ Phép toán:

1. Bỏ sung vào ngăn xếp:

- Vào: T, x.
- Ra: Không có.

{ Thủ tục này bỏ sung phần tử x vào ngăn xếp T được lưu trữ bằng CTLT phân tán }

Procedure Push(Var T; x)

1. Tạo nút mới

$N \leq \text{AVAIL};$

$\text{Infor}(N) := x;$

$\text{Link}(N) := \emptyset;$

2. Nối nút mới vào trên nút T

$\text{Link}(N) := T;$

3. Cho T trở đến nút mới

$T := N;$

Return

2. Loại bỏ nút đỉnh khỏi ngăn xếp:

- Vào: T.
- Ra: Phần tử loại bỏ.

{Hàm này thực hiện việc loại bỏ phần tử ở đỉnh ngăn xếp T lưu trữ bằng CTLT phân tán và trả về phần tử này}

Function Pop(Var T)

1. {Kiểm tra ngăn xếp rỗng}

If $T = \emptyset$ then

begin

write('Ngăn xếp đã rỗng!');

return;

end;

2. {Giữ lại giá trị phần tử đỉnh}

Tg:=Infor(T);

3. {Chuyển con trỏ và hủy nút lấy ra}

P:=T;

T:=Link(T);

P=>AVAIL;

4. {Trả về phần tử loại bỏ}

Pop:=Tg;

Return

3. Kiểm tra ngăn xếp rỗng:

- Vào: T.
- Ra: True nếu NS rỗng, ngược lại FALSE nếu NS không rỗng.

{Hàm kiểm tra NS T lưu trữ bằng CTLT phân tán trả về True nếu NS rỗng, ngược lại FALSE nếu NS không rỗng}

Function IsEmpty(T)

If $T = \emptyset$ then

IsEmpty:=TRUE;

Else IsEmpty:=FALSE;

Return

5. Cài đặt Hàng đợi bằng CTLT phân tán.

☒ Đặc điểm tổ chức:

Đã trình bày trong chương 2!

☑ Cấu trúc lưu trữ:

- Trong CTLT phân tán, các phần tử của hàng đợi được lưu trữ trong các nút nhớ nằm rải rác khắp nơi trong bộ nhớ, những có liên kết với nhau về địa chỉ.
- Mỗi nút nhớ gồm có 2 trường:
 - + Trường Infor: chứa thông tin phần tử dữ liệu.
 - + Trường Link: chứa địa chỉ của nút đứng sau.
 - + Vẽ hình hàng đợi lưu trữ bằng CTLT phân tán
- Nút cuối cùng không có nút đứng sau nên trường $Link = \emptyset$.
- Dùng con trỏ F trỏ đến nút đầu hàng.
Dùng con trỏ R trỏ đến nút cuối hàng.
- Khi loại bỏ một phần tử ra khỏi hàng đợi thì loại bỏ ở lồi trước, do đó F phải trỏ đến nút tiếp theo.
- Khi bổ sung một phần tử vào hàng đợi thì bổ sung ở lồi sau, do đó phải tìm đến nút cuối cùng rồi thêm một nút cuối cùng.
- Khi hàng đợi rỗng thì $F=R=\emptyset$.

☑ Phép toán:

1. Bổ sung vào hàng đợi:

- Vào: Phần tử dữ liệu x, hàng đợi (F, R).
- Ra: Không có.

{ Thủ tục này bổ sung phần tử x vào lồi sau của hàng đợi (F, R) lưu trữ bằng DSLKD }

Procedure QInsert(F, R, x)

1. { Tạo nút mới }

$N \leq AVAIL;$

$Infor(N) := x;$

Link(N):= \emptyset ;

2. {Bổ sung nút mới vào hàng đợi}

If $F=R=\emptyset$ then $F:=R:=N$

Else

begin

Link(R):=N;

R:=N;

end;

Return

2. Loại bỏ phần tử ra khỏi hàng đợi:

- Vào: Hàng đợi (F, R).
- Ra: Giá trị phần tử loại bỏ.

{Thủ tục này bổ sung phần tử x vào lối sau của hàng đợi (F, R) lưu trữ bằng DSLKD}

Function QDelete(F, R)

1. {Kiểm tra hàng đợi rỗng}

If $F=\emptyset$ then

begin

write('Hàng đợi đã rỗng');

return;

end;

2. {Giữ lại phần tử loại bỏ và địa chỉ nút F}

$Tg := \text{Infor}(F);$

$P := F;$

3. {Thay đổi F}

If $F=R$ then $F:=R:=\emptyset$

Else $F:=\text{Link}(F);$

4. {Hủy nút loại bỏ}

$P \Rightarrow \text{AVAIL};$

5. {Trả về phần tử loại bỏ}

$QDelete := Tg;$

Return

3. Kiểm tra hàng đợi rỗng:

- Vào: Hàng đợi (F, R).
- Ra: True nếu hàng đợi rỗng hoặc False nếu hàng đợi không rỗng.

{Hàm này kiểm tra hàng đợi lưu trữ bằng DSLKĐ trả về True nếu hàng đợi rỗng hoặc False nếu hàng đợi không rỗng}

Function IsQEmpty(F, R)

If $F=\emptyset$ then IsQEmpty:=True

Else IsQEmpty:=False;

Return

Câu hỏi ôn tập chương 3:

Câu 19: Trình bày đặc điểm tổ chức của danh sách liên kết đơn (kép).

Câu 20: Trình bày cấu trúc lưu trữ của danh sách liên kết đơn (kép).

Câu 21: Viết giả mã 1 trong các phép toán tùy đề bài.

Câu 22: Trình bày cấu trúc lưu trữ phân tán.

Câu 23: Trình bày cấu trúc lưu trữ của hàng đợi, ngăn xếp theo kiểu phân tán (danh sách liên kết đơn, kép).