

CHƯƠNG 3

DANH SÁCH LIÊN KẾT

1. Giới thiệu về danh sách liên kết
2. Danh sách liên kết đơn
3. Danh sách liên kết vòng
4. Danh sách liên kết kép
5. Cài đặt ngăn xếp và hàng đợi bằng cấu trúc lưu trữ phân tán

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.1

1

1. Giới thiệu về danh sách liên kết

- Danh sách liên kết là danh sách tuyến tính khi sử dụng cấu trúc lưu trữ phân tán. Trong danh sách liên kết, nếu giữa các nút nhớ có 1 liên kết thì ta có DSLK đơn, nếu giữa các nút có 2 liên kết thì ta có DSLK kép.

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

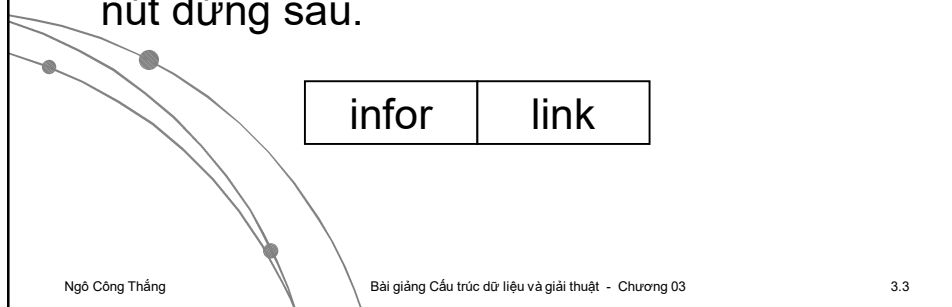
3.2

2

2. Danh sách liên kết đơn

2.1. Quy tắc tổ chức danh sách liên kết đơn

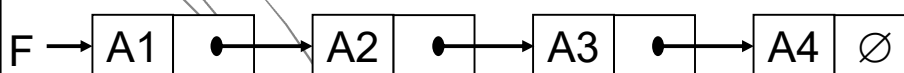
- Trong DSLK đơn, mỗi nút nhớ có cấu trúc gồm hai trường, trường infor chứa phần tử dữ liệu và trường link chứa địa chỉ của nút đứng sau.



3

2.1. Quy tắc tổ chức danh sách liên kết đơn (tiếp)

- Nút cuối cùng trong danh sách không có nút đứng sau nên trường link là rỗng, không chứa địa chỉ, ta ký hiệu là \emptyset .
- Dùng con trỏ F chứa địa chỉ nút đầu tiên để cho phép truy nhập vào tất cả nút trong danh sách.
- Khi danh sách rỗng thì $F = \emptyset$



4

2.1. Quy tắc tổ chức danh sách liên kết đơn (tiếp)

- Để tổ chức lưu trữ một danh sách liên kết thì phải có:
 - Phải có phương tiện chia bộ nhớ ra thành các nút và ở mỗi nút có thể truy nhập vào từng trường.
 - Phải có cơ chế để xác định một nút đang được sử dụng hoặc chưa được sử dụng (nút trống).
 - Phải có cơ chế cung cấp các nút trống khi có yêu cầu sử dụng và thu hồi lại các nút khi không cần dùng nữa.
- Ta ký hiệu:
 - $P \Leftarrow AVAIL$ là phép lấy ra một nút trống có địa chỉ là P (cấp phát một nút)
 - $P \Rightarrow AVAIL$ là phép thu hồi một nút có địa chỉ là P

5

2.2. Một số phép toán trên danh sách liên kết đơn

- Ký hiệu: Một nút có địa chỉ là p (được trỏ bởi p) thì Infor(p) và Link(p) tương ứng chỉ trường Infor và Link của nút đó.

a) Bổ sung một nút mới vào danh sách

Cho danh sách liên kết đơn F, M là con trỏ trỏ tới một nút trong danh sách. Viết thủ tục bổ sung phần tử dữ liệu x **vào sau** nút M.

6

PP chung bổ sung vào cấu trúc lưu trữ phân tán bất kể cấu trúc dữ liệu là gì

- B1: Tạo nút nhớ mới, đưa phần tử dữ liệu vào nút nhớ và cho các trường địa chỉ bằng rỗng.
- B2: Nối nút nhớ mới vào trong cấu trúc sao cho vẫn đảm bảo liên kết của cấu trúc
=> Thay đổi liên kết theo quy tắc thay đổi liên kết.

7

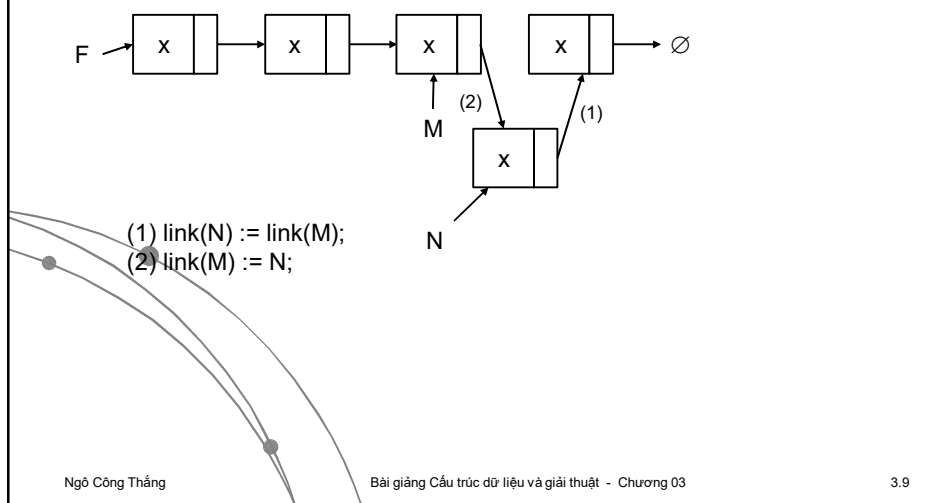
Quy tắc thay đổi liên kết trong cấu trúc lưu trữ phân tán

- Luôn phải xét trường hợp cấu trúc rỗng
- Xem trường hợp nào làm thay đổi biến của cấu trúc thì phải xét xử lý riêng, còn lại thì xử lý chung.

Ghi nhớ: Khi làm về cấu trúc lưu trữ phân tán thì luôn phải vẽ hình, thực hiện thao tác trên hình, đánh số thể hiện thứ tự thực hiện rồi sau đó chuyển thành các lệnh giả mã.

8

BỔ sung vào sau nút M trong DSLKD F



9

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

a) Bổ sung một nút mới vào danh sách:

- Vào: F, M, x

- Ra: Không có

{Thủ tục này bổ sung phần tử dữ liệu x **vào sau** nút trở bởi M trong danh sách liên kết đơn F}

Procedure SLPostInsert(Var F; M, x)

1. {Tạo nút mới}

N ← AVAIL;

infor(N):=x; link(N):= ∅;

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.10

10

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

2. {Nối nút mới vào sau nút M}

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

Else begin

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

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

end;

Return

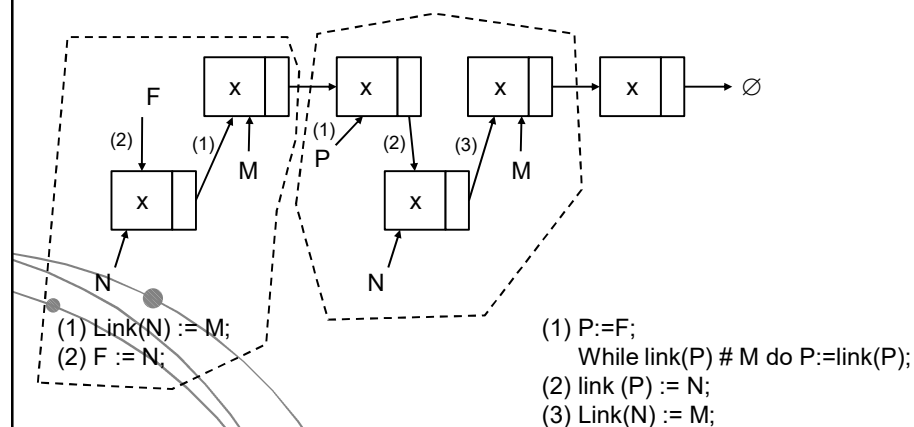
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.11

11

Bổ sung vào trước nút M trong DSLKD



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.12

12

PP chung loại bỏ phần tử khỏi cấu trúc lưu trữ phân tán bất kể cấu trúc dữ liệu là gì

- B1: Ngắt liên kết với nút cần loại bỏ sao cho vẫn đảm bảo liên kết của cấu trúc => Thay đổi liên kết theo các quy tắc thay đổi liên kết.
- B2: Hủy nút chứa phần tử cần loại bỏ

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.13

13

Quy tắc thay đổi liên kết trong cấu trúc lưu trữ phân tán

- Luôn phải xét trường hợp cấu trúc rỗng
- Xem trường hợp nào làm thay đổi biến của cấu trúc thì phải xét xử lý riêng, còn lại thì xử lý chung.

Ghi nhớ: Khi làm về cấu trúc lưu trữ phân tán thì luôn phải vẽ hình, thực hiện thao tác trên hình, đánh số thể hiện thứ tự thực hiện rồi sau đó chuyển thành các lệnh giả mã.

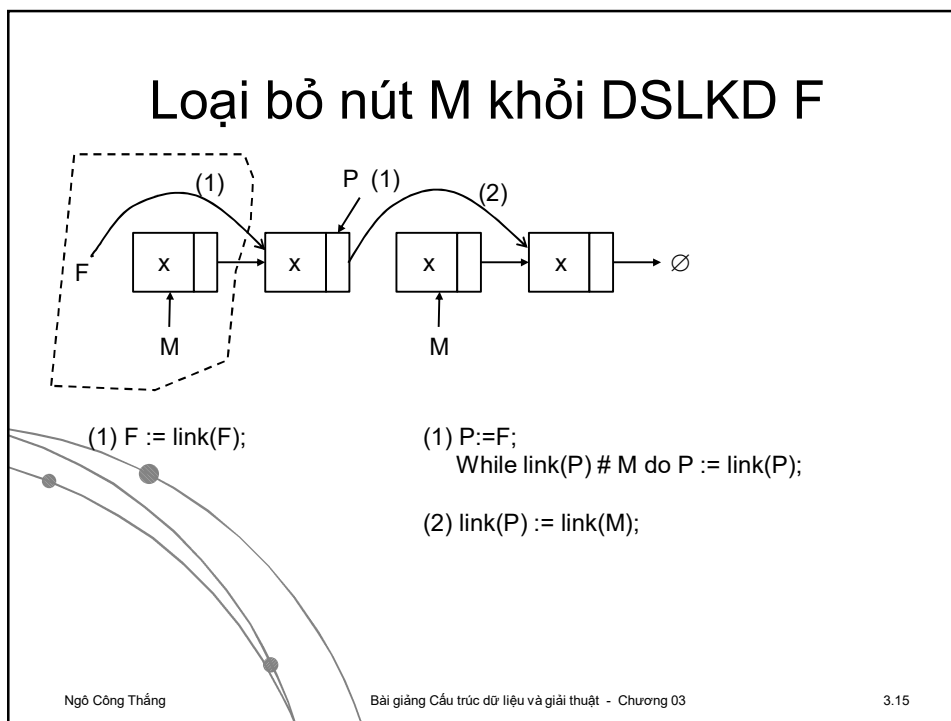
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.14

14

Loại bỏ nút M khỏi DSLKD F



15

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

b) Loại bỏ một nút khỏi danh sách

- Vào: F, M

- Ra: Không

{Thủ tục này loại bỏ nút trỏ bởi M khỏi danh sách liên kết đơn F}

Procedure SLDelete(Var F; M)

1. { Trường hợp danh sách rỗng }

If $F = \emptyset$ then begin

Write('danh sách rỗng')

Return

end

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.16

16

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

```

2. {Ngắt kết nối với nút M}
   {M là nút đầu tiên của danh sách}
   If M=F then F:=link(F) Else
   begin
       {Tìm đến nút đứng trước nút M }
       P:=F;
       ● While link(P) # M do P:=link(P);
       ● {Nối nút trước M với nút sau M}
       link(P):=link(M);
   end;

```

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.17

17

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

```

3. {Hủy nút M}
   M ⇒ AVAIL;
Return

```

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.18

18

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

c) Duyệt danh sách

- Vào: F

- Ra: Không

{Thủ tục này duyệt danh sách liên kết đơn F và đưa ra phần tử dữ liệu trong các nút của ds}

Procedure SLDisplay(F)

1) P := F;

2) While P # \emptyset do

begin

Write(infor(P)); P := link(P);

end;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.19

19

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

d) Ghép hai danh sách liên kết đơn

Cho 2 danh sách liên kết đơn P và Q, viết thủ tục nối Q vào sau P, P trở tới DSLKD sau khi nối.

Procedure SLConcat(Var P; Q)

1. {Danh sách trở bởi q rỗng}

If Q = \emptyset then Return

2. {Trường hợp danh sách trở bởi p rỗng}

If P = \emptyset then begin

P:=Q

return

end

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.20

20

2.2. Một số phép toán trên danh sách liên kết đơn (tiếp)

d) Ghép hai danh sách liên kết đơn

3. {Tìm đến nút cuối danh sách P}

P1:= P

While link(P1) # \emptyset do P1:=link(P1);

4. {Nối Q vào sau nút cuối của P}

Link(P1):=Q;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.21

21

Ưu nhược điểm của danh sách liên kết đơn

- Với danh sách tuyến tính động, trong quá trình xử lý luôn có bổ sung, loại bỏ thì tổ chức danh sách liên kết là hợp lý, tận dụng được các vùng nhớ nằm rải rác trong bộ nhớ.
- Chỉ có phần tử đầu tiên là truy nhập ngay được, các phần tử khác phải truy nhập qua phần tử đứng trước nó.
- Tốn bộ nhớ do phải lưu cả 2 trường infor và link ở mỗi nút.

Ngô Công Thắng

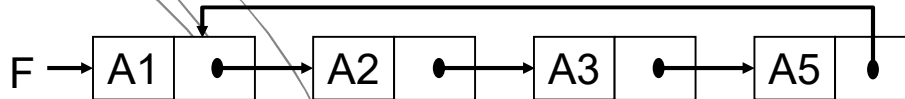
Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.22

22

3. Danh sách liên kết vòng

- Danh sách liên kết vòng (Circularly Linked List) là một dạng cải tiến của danh sách liên kết đơn.
- Trong danh sách liên kết vòng, trường địa chỉ của nút cuối cùng không phải là rỗng mà lại chứa địa chỉ của nút đầu tiên của danh sách.



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.23

23

3. Danh sách liên kết vòng (tiếp)

- Ưu nhược điểm của danh sách nối vòng:
 - Danh sách nối vòng làm cho việc truy nhập vào các nút trong danh sách linh hoạt hơn. Ta có thể truy nhập vào danh sách bắt đầu từ một nút nào cũng được, không nhất thiết phải từ nút đầu tiên. Nút nào cũng có thể là nút đầu tiên và con trỏ F trở vào nút nào cũng được.
 - Nhược điểm của danh sách nối vòng là trong xử lý nếu không cẩn thận sẽ dẫn tới một chu trình không kết thúc.

Ngô Công Thắng

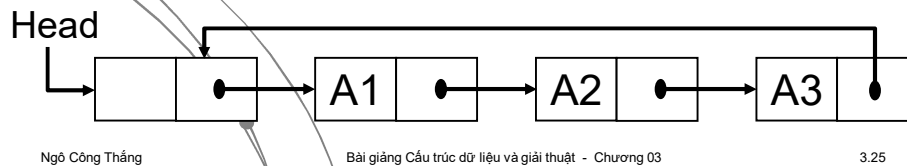
Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.24

24

3. Danh sách liên kết vòng (tiếp)

- Để khắc phục nhược điểm của danh sách nối vòng ta đưa thêm vào một nút đặc biệt gọi là “nút đầu danh sách” (list head node). Trường Infor của nút này không chứa dữ liệu, con trỏ HEAD trỏ tới nút đầu danh sách này cho phép ta truy nhập vào danh sách.



Ngô Công Thắng

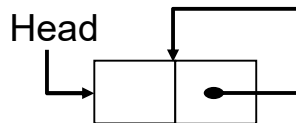
Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.25

25

3. Danh sách liên kết vòng (tiếp)

- Việc dùng thêm nút đầu danh sách đã làm cho danh sách luôn có ít nhất 1 nút nên không bao giờ rỗng. Danh sách có 1 nút HEAD có $LINK(Head) = Head$.



- Các phép toán bổ sung và loại bỏ nút trong danh sách liên kết vòng tương tự danh sách liên kết đơn.

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03



3.26

26

Bài tập (ctdlgt-dslkd.cpp)

- Cài đặt cấu trúc dữ liệu danh sách liên kết đơn (DSLKD) có phần tử là số nguyên với bốn phép toán:
 - 1) Bổ sung phần tử x vào sau nút M
 - 2) Loại bỏ nút M
 - 3) Duyệt DSLKD
 - 4) Ghép hai danh sách

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.27

27

4. Danh sách liên kết kép

4.1. Giới thiệu

- Trong danh sách liên kết kép, mỗi nút nhớ có cấu trúc gồm 3 trường như sau:



infor: Chứa phần tử dữ liệu.

left: Con trỏ trỏ tới nút đứng trước

right: Con trỏ trỏ tới nút đứng sau

Ngô Công Thắng

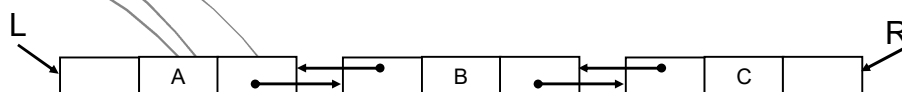
Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.28

28

4.1. Giới thiệu (tiếp)

- left của nút cực trái và right của nút cực phải phải có giá trị là \emptyset .
- Dùng con trỏ L trở vào nút cực trái, con trỏ R trở vào nút cực phải để truy nhập vào danh sách cả 2 chiều.
- Khi danh sách rỗng thì $L = R = \emptyset$.



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.29

29

4.2. Các phép toán trên danh sách liên kết kép

- a) Bổ sung phần tử dữ liệu vào danh sách
- Cho danh sách liên kết kép (L, R). M là con trỏ trở tới một nút trong danh sách. Bổ sung phần tử dữ liệu x **vào trước** nút M.

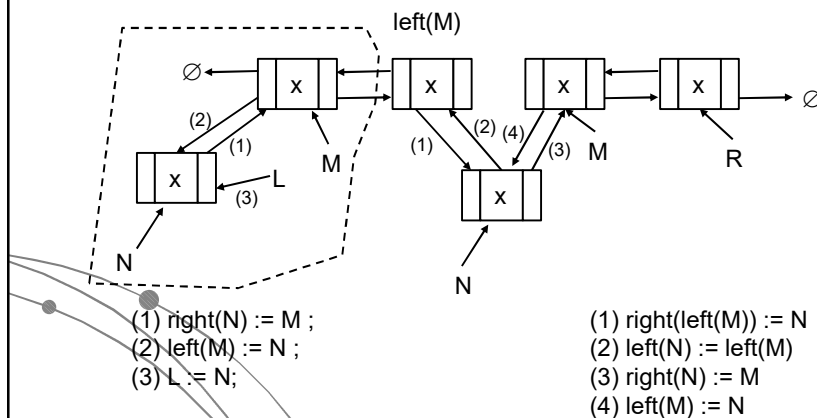
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.30

30

Bổ sung x vào trước nút M trong DSLKK



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.31

31

4.2. Các phép toán trên danh sách liên kết kép

- Vào: (L,R),M,x

- Ra: Không có

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

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

1. {Tạo nút mới}

• N ← AVAIL

infor(N) := x

left(N) := right(N) := ∅

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.32

32

a) Chèn thêm một nút vào danh sách

2. {Trường hợp danh sách rỗng}

If $L=R=\emptyset$ then begin

$L := R := N$;

Return;

end

3. {M trở tới nút cực trái}

If $M=L$ then begin

$right(N) := M$;

$left(M) := N$;

$L := N$;

Return;

end

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.33

33

a) Chèn thêm một nút vào danh sách

4. {Trường hợp còn lại}

$right(left(M)) := N$

$left(N) := left(M)$

$right(N) := M$

$left(M) := N$

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.34

34

b) Loại bỏ một nút ra khỏi danh sách liên kết kép

- Cho danh sách liên kết kép L, R. M là con trỏ trỏ tới một nút trong danh sách. Loại bỏ nút M.

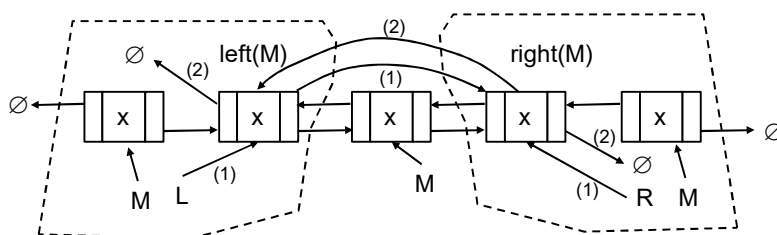
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.35

35

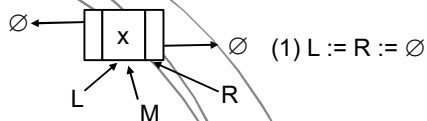
Loại bỏ nút M trong DSLKK



(1) $L := \text{right}(L)$
 (2) $\text{left}(L) := \emptyset;$

(1) $\text{right}(\text{left}(M)) := \text{right}(M)$
 (2) $\text{left}(\text{right}(M)) := \text{left}(M)$

(1) $R := \text{left}(R)$
 (2) $\text{right}(R) := \emptyset;$



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.36

36

b) Loại bỏ một nút ra khỏi danh sách liên kết kép

- Cho danh sách liên kết kép L, R. M là con trỏ trỏ tới một nút trong danh sách cần loại bỏ.

- Vào: (L,R), M

- Ra: Không có

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

Procedure DLDelete(Var L, R; M)

1. {Trường hợp danh sách rỗng }

if L=R=∅ then begin

Write(' danh sach rong ')

Return

end

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.37

37

b) Loại bỏ một nút ra khỏi danh sách liên kết kép

2. {Ngắt kết nối với nút M}

Case

M = L = R: Begin {Danh sach chỉ còn 1 nút M}

L:=R:= ∅;

end

M=L: Begin { Nút cực trái bị loại }

L := right(L)

left(L) := ∅

end

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.38

38

b) Loại bỏ một nút ra khỏi danh sách liên kết kép

```
M=R: Begin { Nút cực phải bị loại }
      R := left(R)
      right(R) := ∅
      end
```

```
ELSE: begin
      right(left(M)):=right(M)
      left(right(M)):=left(M)
      end;
```

End Case

3.{Hủy nút M}

M ⇒ AVAIL;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03



3.39

39

c) Duyệt danh sách liên kết kép và đưa ra các phần tử của danh sách

- Vào: L, R

- Ra: Không có

{Thủ tục này duyệt danh sách từ trái sang phải}

Procedure DLDisplay(L, R);

1) P:= L;

2) While P # ∅ do

begin

Write(infor(P));

P := right(P);

end;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03



3.40

40

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

5.1. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp

- Các phần tử dữ liệu của ngăn xếp được lưu trữ trong các nút nhớ nằm rải rác khắp nơi trong bộ nhớ, mỗi nút nhớ có cấu trúc gồm 2 trường
- Nút dưới cùng (nút đáy) có link bằng \emptyset

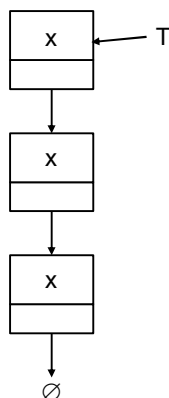
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.41

41

Cấu trúc lưu trữ phân tán của ngăn xếp



Khi ngăn xếp rỗng, $T = \emptyset$

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.42

42

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

- Vào: T, x

- Ra: Không có

{Thủ tục này bổ sung phần tử x vào ngăn xếp T lưu trữ phân tán}

Procedure push(Var T; x)

1) {Tạo nút mới}

N <= AVAIL; infor(N) := x; link(N) := \emptyset ;

2) {Nối nút mới vào trên nút T}

link(N) := T;

3) {Cho T trở tới nút mới}

T := N;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.43

43

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

- Vào: T

- Ra: Phần tử dữ liệu loại bỏ

{Hàm này loại bỏ phần tử đỉnh ngăn xếp T sử dụng cấu trúc lưu trữ phân tán và trả về phần tử này}

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.44

44

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

Function pop(Var T)

1) {Kiểm tra ngăn xếp rỗng}

If T = \emptyset then begin

write('Ngan xep da rong'); return; end;

2) {Giữ lại phần tử đỉnh}

tg := infor(T); P:=T;

3) {Cho T trở xuống nút bên dưới}

T := link(T);

4) {Hủy nút đỉnh và trả về phần tử đỉnh}

P => AVAIL; pop := tg;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.45

45

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

- Vào: T

- Ra: TRUE nếu ngăn xếp rỗng, FALSE nếu không rỗng

{Hàm kiểm tra ngăn xếp T lưu trữ phân tán, trả về TRUE nếu n.xếp rỗng và FALSE nếu chưa rỗng}

Function isEmpty(T)

If T = \emptyset then isEmpty:=TRUE;

Else isEmpty:=FALSE;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.46

46

5. Sử dụng cấu trúc lưu trữ phân tán cho ngăn xếp và hàng đợi

- Vào: T

- Ra: Phần tử dữ liệu đỉnh ngăn xếp

{Hàm này trả về phần tử đỉnh ngăn xếp T lưu trữ phân tán}

Function top(T)

1) {Kiểm tra ngăn xếp rỗng}

If T = \emptyset then begin

write('Ngan xep da rong'); return; end;

2) {Trả về phần tử đỉnh}

top := infor(T);

Return

Ngô Công Thắng

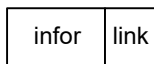
Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.47

47

5.2. Sử dụng cấu trúc lưu trữ phân tán cho hàng đợi

- Trong cấu trúc lưu trữ phân tán, các phần tử dữ liệu của hàng đợi được lưu trữ trong các nút nhớ, mỗi nút nhớ có cấu trúc gồm 2 trường, trường infor chứa phần tử dữ liệu, trường link chứa địa chỉ của nút đứng sau.



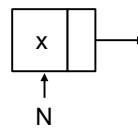
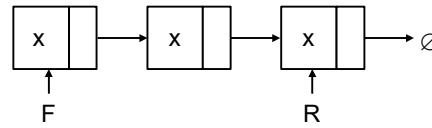
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.48

48

Cấu trúc lưu trữ phân tán của hàng đợi



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.49

49

5.2. Sử dụng cấu trúc lưu trữ phân tán cho hàng đợi

- Vào: (F, R), x

- Ra: Không có

{Thủ tục này bổ sung phần tử x vào lối sau của hàng đợi (F, R) sử dụng cấu trúc lưu trữ phân tán}

Procedure QInsert(Var F,R; x)

1) {Tạo nút mới}

N <= AVAIL;

infor(N) := x; link(N) := ∅;

2) {Nối nút mới vào sau R}

If F=R=∅ Then F:=N Else link(R) := N;

3) {Cho R trở tới nút mới}

R := N;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.50

50

5.2. Sử dụng cấu trúc lưu trữ phân tán cho hàng đợi

- Vào: (F, R)

- Ra: Phần tử dữ liệu loại bỏ

{Hàm này loại bỏ phần tử ở lối trước của hàng đợi (F, R) lưu trữ phân tán và trả về phần tử loại bỏ}

Function QDelete(Var F, R)

1){Kiểm tra hàng đợi rỗng}

If F=R=∅ then begin

write('Hàng đợi đã rỗng.');

return;

end;

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.51

51

5.2. Sử dụng cấu trúc lưu trữ phân tán cho hàng đợi

2) {Giữ lại nút lối trước (nút đầu hàng)}

tg := infor(F); P := F;

3) {Cho F trở tới nút đứng sau}

If F=R then F:=R:=∅

Else F := link(F);

4) {Hủy nút và trả về phần tử dữ liệu}

P => AVAIL;

QDelete := tg;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.52

52

5.2. Sử dụng cấu trúc lưu trữ phân tán cho hàng đợi

- Vào: (F, R)

- Ra: True - rỗng, False - không rỗng

{Hàm này kiểm tra hàng đợi rỗng}

Function QIsEmpty(F, R)

If F=R= \emptyset then QIsEmpty := True

Else QIsEmpty := False;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.53

53

Bài tập

- (ctdlgt-nganxep2.cpp) Cài đặt cấu trúc dữ liệu ngăn xếp lưu trữ phân tán có phần tử là ký tự. Ứng dụng chuyển số nguyên hệ 10 sang hệ 16.

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.54

54