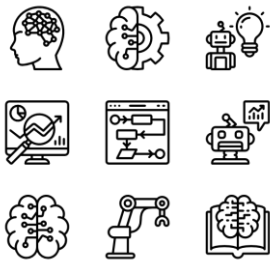


Computer Science for Practicing Engineers

Danh sách liên kết (Linked List)



TS. Huỳnh Bá Diệu
Email: dieuhb@gmail.com
Phone: 0914146868

1

Danh sách liên kết

Nội dung

1. Kiểu danh sách liên kết
2. Các loại danh sách liên kết
3. Các thao tác trên kiểu dữ liệu danh sách
4. Sử dụng kiểu danh sách trên các ngôn ngữ lập trình

2

Các cấu trúc dữ liệu trừu tượng ADT (Abstract Data Type)

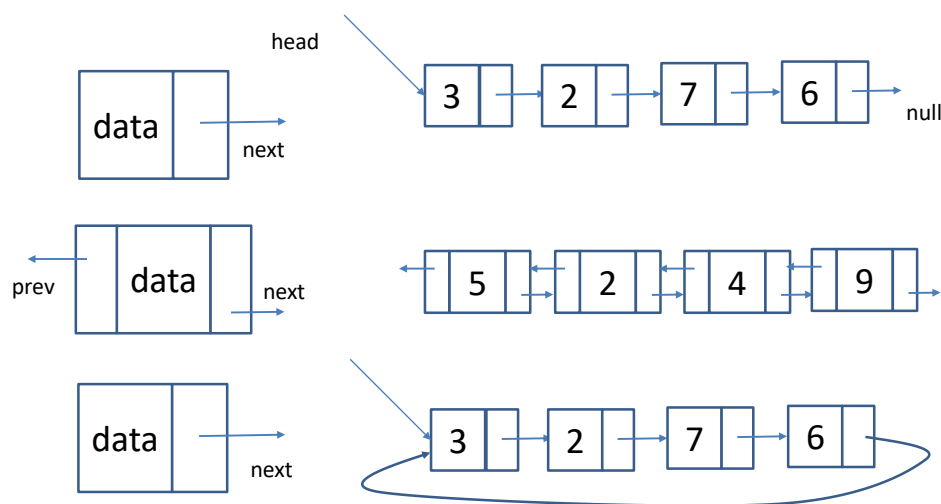
- Danh sách liên kết (Đơn, Đôi, Vòng)
- Ngăn xếp (Stack)
- Hàng đợi (Queue)
- Cây (Tree)
- Bảng băm (Hash Table)

Mỗi ngôn ngữ lập trình đều có các kiểu dữ liệu nguyên thủy (cơ sở). Từ kiểu dữ liệu nguyên thủy này ta có thể định nghĩa các kiểu dữ liệu mới.

Các kiểu dữ liệu nguyên thủy thường là: ký tự, số, chuỗi, mảng..

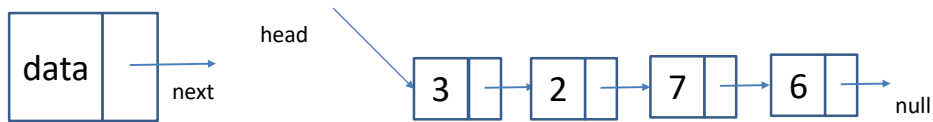
3

Các loại danh sách liên kết (Linked List)



4

Khai báo danh sách liên kết (Linked List)



```

class Node
{
    int data;
    Node next;
    .....
}
  
```

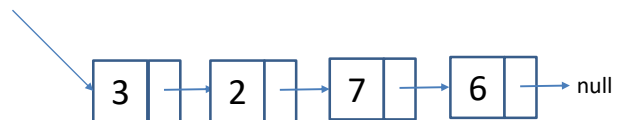
```

typedef struct Node
{
    int data;
    Node *next;
};
  
```

5

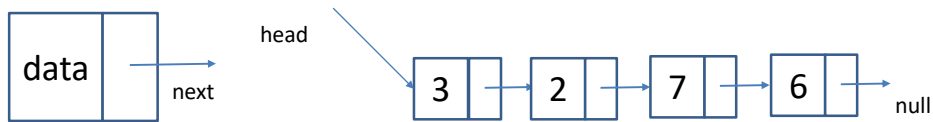
Các thao tác trên danh sách liên kết đơn

- Tạo danh sách
- Duyệt danh sách
- Thêm phần tử vào đầu danh sách
- Thêm phần tử vào cuối danh sách
- Thêm vào vị trí k
- Xóa phần tử đầu danh sách
- Xóa phần tử cuối danh sách
- Xóa phần tử vị trí k
- Sắp xếp
- Đảo danh sách



6

Khai báo một phần tử trong Danh sách liên kết



```

class Node
{
    int data;
    Node next;
    Node(int x) { data =x; next = null;} // tạo nốt có giá trị x
    Node(int x, Node t) { data =x; next = t;} // tạo nốt đứng trước nốt t
}

```

7

Danh sách liên kết đơn (Linked List)

```

public class SList
{
    Node head;
    void tao() { } // tạo danh sách gồm các số, dừng khi nhập số 0
    void in() { } // in nội dung danh sách
    // sẽ bổ sung các phương thức khác tại đây
    public static void main(String argv)
    {
        SList L = new SList ();
        L.tao(); L.in();
    }
}

```

8

Nhập danh sách liên kết (Linked List)

```
void tao() {
    head= null;
    Node cuoi= null;
    Scanner kb= new Scanner(System.in);
    while (true)
    {
        System.out.print(" nhap 1 so them vao ds (0 dung):");
        int x= kb.nextInt();    if(x==0) break;        Node t= new Node(x);
        if(head==null) head= cuoi=t;    else {cuoi.next=t; cuoi=t;}
    }
}
```

9

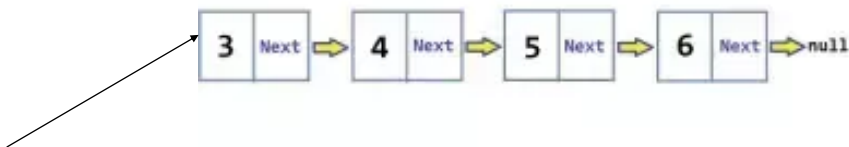
In danh sách liên kết (Linked List)

```
void in() {
    System.out.print("\n Noi dung danh sach:\n ==>");
    Node p= head;
    while (p!=null)
    {
        System.out.print(p.data + "-->");
        p=p.next;
    }
    System.out.print(" null\n");
}
```

10

Tính tổng các phần tử trong danh sách liên kết

Cách làm như thế nào???



11

Tính tổng các phần tử trong danh sách liên kết

Tính tổng các nốt trong danh sách liên kết

Thuật toán:

- Cho p trỏ đầu danh sách, khởi gán k=0
- Trong khi p khác rỗng
 - + $k = k + p.data$;
 - + $p = p.next$
- Trả về giá trị của k

12

Danh sách liên kết (Linked List)

```
int tong()
{
    int k=0;
    Node p= head;
    while(p!= null)
        { k= k+p.data; p= p.next;}
    return k;
}
```

```
int tongdq(Node L)
{
    if(L==null) return 0;
    else return L.data + tongdq(L.next);
}

int tongdq()
{
    return tongdq(head);
}
```

S.O.P (" Tong cac phan tu =" + L.tongdq());

13

Danh sách liên kết (Linked List)

Thực hiện theo phương pháp lặp và đệ quy (10 phút)

Đếm số phần tử lẻ trong danh sách?

Kiểm tra trong danh sách có chứa giá trị x hay không?

Kiểm tra danh sách có tăng dần hay không?

Đếm số phần tử có giá trị là bội của 3 trong danh sách?

14

Chèn giá trị x vào cuối danh sách liên kết

Các trường hợp có thể có???



15

Chèn giá trị x vào cuối danh sách liên kết

Thuật toán

Nếu danh sách rỗng thì tạo nốt mới và gán head = nốt vừa tạo

Ngược lại

- + Cho p trở đến nốt cuối của danh sách
- + Tạo 1 nốt mới t có dữ liệu là x và liên kết là null
- + Nối nốt mới sau nốt cuối ($p.next = t$)

16

Chèn giá trị x vào đầu danh sách liên kết

```
void chend(int x) { head = new Node(x, head); }
void chenc(int x)
{
    if (head == null) head = new Node(x);
    else
    {
        // den not cuoi va tao mot not noi sau not cuoi
        Node p = head; while (p.next != null) p = p.next;
        Node t = new Node(x);    p.next = t;
    }
}
```

17

Chèn giá trị x vào vị trí k trong danh sách liên kết

Các trường hợp có thể có?



18

Chèn giá trị x vào vị trí k trong danh sách liên kết

Thuật toán

Nếu $k < 1$ thì báo không chèn được

Ngược lại

 nếu $k = 1$ thì chèn đầu danh sách

 ngược lại

 {

 Cho p trở đến nốt k-1 của danh sách

 Nếu p bằng rỗng thì báo vị trí chèn không hợp lệ

 Ngược lại tạo nốt mới t và gắn t nằm giữa p và p.next

 }

19

Chèn giá trị x vào vị trí k trong danh sách liên kết

```
void chenk(int x, int k)
```

```
{
```

```
    if(k<1) System.out.println("\n Vi tri chen khong hop le!");
```

```
    else    if (k==1) head= new Node(x, head);
```

```
        else
```

```
        {
```

```
            // bo sung vao day
```

```
        }
```

```
}
```

20

Chèn giá trị x vào vị trí k trong danh sách liên kết

```
void chenK(int x, int k) {
    if(k<1) System.out.println("\n Vị trí chèn không hợp lệ!");
    else if (k==1) head= new Node(x, head);
    else {
        // cho p trỏ đến vị trí k-1
        int vt=1;        Node p= head;
        while(p!= null && vt<k-1) { vt++; p=p.next;}
        // Kiểm tra và nối vào
        if(p==null) System.out.println("\n Vị trí chèn không hợp lệ!");
        else { Node t= new Node(x, p.next); p.next=t; }
    }
}
```

21

Xoá nốt trong danh sách liên kết (Linked List)

Xoá nốt trong danh sách liên kết

Xoá nốt đầu, xoá nốt cuối, xoá nốt thứ k

Khi xoá một phần tử thuộc danh sách thì yêu cầu trước tiên là danh sách không phải là danh sách rỗng.

```
void xoa()
{
    if(head==null) SOP("danh sách rỗng");
    else head= head.next;
}
```

22

Xoá nốt thứ k trong danh sách liên kết

Các trường hợp có thể có???



VectorStock.com/6418811

23

Xoá nốt thứ k trong danh sách liên kết

Thuật toán

Nếu $k < 1$ || danh sách rỗng thì báo không xoá được

Ngược lại :

Nếu $k=1$ thì xoá nốt đầu tiên

ngược lại:

{

Cho p trở đến nốt thứ $k-1$;

Nếu p rỗng hoặc $p.next = \text{rỗng}$ thì báo không xoá được

ngược lại cho $p.next = p.next.next$ (bỏ qua nốt ở vị trí $p.next$)

}

24

Xoá nốt thứ k trong danh sách liên kết

```
void xoak(int k) {
    if(k<1 || head==null) System.out.println("\n Vi tri xoa khong hop le");
    else if(k==1) head= head.next;
    else {
        // cho p tro den vi tri k-1
        int vt=1; Node p= head; while(p!= null&& vt<k-1) { vt++ ; p=p.next;}
        // kiem tra co not o vi tri k khong
        if(p==null || p.next==null) System.out.println("\n Vi tri xoa khong hop le!");
        else p.next= p.next.next;
    }
}
```

25

Chèn nốt vào vị trí k trong danh sách liên kết (ĐQ)

Thuật toán: **Node Chenk_dq(Node L, int x, int k)**

- Nếu k==1 thì chèn đầu danh sách: L= new Node(x,L);

Ngược lại

Nếu k>1 và L == null thì báo ko chèn được

Ngược lại: chèn vào vị trí thứ k-1 của danh sách kế tiếp

// L.next = Chenk_dq(L.next, x, k-1)

- return L;

void chenk dq (int x, int k) { head= Chenk_dq(head, x, k); }

26

Chèn nốt vào vị trí k trong danh sách liên kết (ĐQ)

Thuật toán:

Node Chenk_dq(Node L, int x, int k)

```
{
    if(k==1) L= new Node(x, L);
    else if(k>1 && L==NULL) S.O.P ("Khong chen duoc");
        else L.next= Chenk_dq(L.next, x, k-1 );
    return L;
}
void chenkdq (int x, int k) { head= Chenk_dq(head, x, k); }
```

27

Xóa nốt ở vị trí k trong danh sách liên kết (ĐQ)

Thuật toán: Xoak_dq(Node L, int k)

- Nếu k < 1 hoặc danh sách rỗng thì báo ko xóa được

Ngược lại

Nếu k=1 thì xóa nốt trở bởi L (L=L.next)

Ngược lại : xóa nốt thứ k-1 của danh sách kế tiếp sau L

// L.next = Xoak_dq(L.next, k-1)

- return L;

28

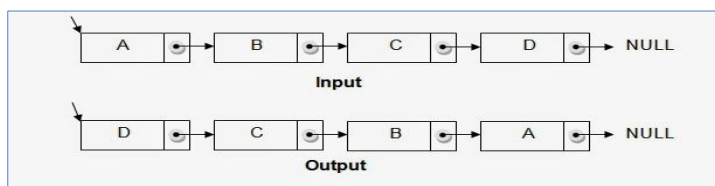
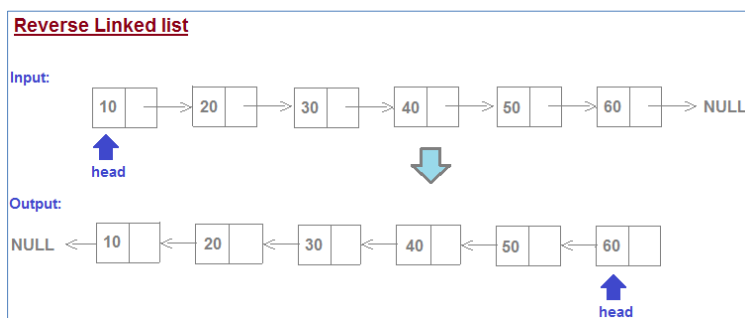
Xóa nốt ở vị trí k trong danh sách liên kết (ĐQ)

Thuật toán

```
Node Xoak_dq(Node L, int k )
{
    if(L==null) S.O.P(" Ko xoa duoc");
    else if(k==1) L= L.next;
        else L.next = Xoak_dq( L.next, k-1 );
    return L;
}
```

29

Đảo danh sách liên kết

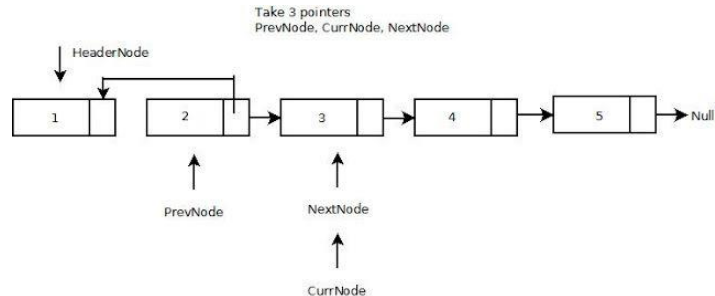


30

Đảo danh sách liên kết

Ví dụ danh sách có 5 phần tử

Đảo từ 1 đến 1,
Đảo từ 2 đến 1
Đảo từ 3 đến 1
Đảo từ 4 đến 1
Đảo từ 5 đến 1



Khi đảo từ 5 đến 1 ta đảo từ 1 đến 4, sau đó cho nốt 5 chỉ đến phần đảo từ 1 đến 4 và cập nhật đầu danh sách là nốt thứ 5

Tương tự khi đảo từ 4 đến 1, ta thực hiện đảo từ 1 đến 3 sau đó cho nốt kế tiếp nốt 4 là phần đảo của danh sách từ 1 đến 3

31

Đảo danh sách liên kết

Thuật toán

Khi đảo từ 5 đến 1 ta đảo từ 1 đến 4, sau đó cho nốt 5 chỉ đến phần đảo từ 1 đến 4 và cập nhật đầu danh sách là nốt thứ 5

Tương tự khi đảo từ 4 đến 1, ta thực hiện đảo từ 1 đến 3 sau đó cho nốt kế tiếp nốt 4 là phần đảo của danh sách từ 1 đến 3

Như vậy ta cần:

- 1 con trỏ p lưu trữ nốt đang xét
- 1 con trỏ q lưu trữ nốt phía sau nốt đang xét
- 1 con trỏ lưu trữ t phần đảo phía trước nốt đang xét (ban đầu t= rỗng)

32

Đảo danh sách liên kết

Thuật toán:

- Node t= rỗng
- Node p= head;
- Trong khi p <> rỗng
 - + Node q= p.next; // lưu trữ nốt phía sau
 - + p.next = t; cho nốt kế tiếp của p là phần đảo phía trước
 - + t = p; // cập nhật phần đảo từ nốt đầu tiên đến nốt p
 - + p= q; // cho p trở đến nốt kế tiếp trong danh sách để đảo
- head= t; // cập nhật lại đầu danh sách (là phần đảo)

33

Đảo danh sách liên kết

CODE hoàn chỉnh như thế nào???



34

Bài tập về nhà

1. Hoán vị nốt thứ i và j trong danh sách (chỉ thay đổi liên kết)
2. Xóa các số lẻ trong danh sách
3. Sắp xếp danh sách tăng dần
4. Tách danh sách thành hai danh sách, 1 danh sách chứa các số lẻ, 1 danh sách chứa số chẵn
5. Kiểm tra danh sách chứa quá 3 số lẻ không

ÁP DỤNG CÁC BÀI TẬP TRÊN DANH SÁCH LIÊN KẾT ĐÔI !

Bài làm trên vở, Không THÌ MÀ LÀ NHƯNG DO!!!

35

Sử dụng Kiểu danh sách liên kết trong java



The screenshot shows the Java API documentation for the `java.util.LinkedList` class. The navigation bar at the top includes links for Overview, Package, Class (highlighted), Use, Tree, Deprecated, Index, and Help. Below the navigation bar, there are links for Prev Class, Next Class, Frames, No Frames, and All Classes. The main content area shows the class hierarchy: `java.lang.Object` is the superclass, followed by `java.util.AbstractCollection<E>`, `java.util.AbstractList<E>`, `java.util.AbstractSequentialList<E>`, and finally `java.util.LinkedList<E>`. The 'Type Parameters' section indicates that `E` is the type of elements held in this collection. The 'All Implemented Interfaces' section is also visible.

36

Sử dụng Kiểu danh sách liên kết trong java

SN	Methods	
1	void add(int index, Object o)	Thêm vào vị trí i đối tượng o
2	boolean add(Object o)	Thêm đối tượng o vào danh sách
3	boolean addAll(Collection c)	
4	boolean addAll(int index, Collection c)	
5	void addFirst(Object o)	
6	void addLast(Object o)	
7	void clear()	
8	Object clone()	
9	boolean contains(Object o)	Kiểm tra danh sách có chứa đối tượng o ko?
10	Object get(int index)	Cho lại đối tượng ở vị trí i
11	Object getFirst()	

37

Sử dụng Kiểu danh sách liên kết trong java

SN	Methods	
12	Object getLast()	
13	int indexOf(Object o)	Cho lại vị trí của đối tượng o trong danh sách
14	int lastIndexOf(Object o)	
15	Object remove(int index)	Xóa phần tử thứ i
16	boolean remove(Object o)	Xóa đối tượng o
17	Object removeFirst()	
18	Object removeLast()	
19	Object set(int index, Object element)	Cho vị trí phần tử ở vị trí thứ i thành giá trị o
20	int size()	Cho lại số lượng phần tử trong danh sách
21	...	

38

Sử dụng Kiểu danh sách liên kết trong java

```
import java.util.LinkedList;
public class LExample {
public static void main(String args[]) {
    LinkedList<String> L = new LinkedList<String>();
    L.add("Item1"); L.add("Item2"); L.add("Item3");
    System.out.println("Linked List Content: " +L);
    L.addFirst("First Item");
    L.addLast("Last Item");
    System.out.println("Linked List Content : " +L);
    Object firstvar = L.get(0);
    System.out.println("First element: " +firstvar);
}
```

39

Sử dụng Kiểu danh sách liên kết trong java

```
import java.util.L;
public class LExample {
public static void main(String args[]) {
    L<String> L = new L<String>();
    L.add("Item1"); L.add("Item2"); L.add("Item3");
    System.out.println("Linked List Content: " +L);
    L.addFirst("First Item"); L.addLast("Last Item");
    System.out.println("L Content after addition: " +L);
    Object firstvar = L.get(0);
    System.out.println("First element: " +firstvar);
    L.set(0, "Changed first item"); Object firstvar2 = L.get(0);
    System.out.println("First element after update by set method: " +firstvar2);
    L.removeFirst(); L.removeLast();
    System.out.println("L after deletion : " +L);
    L.add(0, "Newly added item"); L.remove(2);
    System.out.println("Final Content: " +L); }
}
```

40

Sử dụng Kiểu danh sách liên kết trong java

Bài tập:

Cho file LIST.inp chứa các từ. Dùng kiểu LinkedList của Java viết chương trình:

- Đọc các từ trong file dữ liệu sau đó cho vào danh sách liên kết. Các từ đã có thì không đưa vào danh sách.
- Ghi các từ trong danh sách ra file LIST.out

Tên chương trình: List1.java

41

Đọc ghi file trong java

Làm thế nào để đọc file???

Làm thế nào để ghi file???

42

Sử dụng Kiểu danh sách liên kết trong java

```
Scanner kb = new Scanner(new File("List.inp"));
String x= kb.next(); // đọc 1 từ
```

```
PrintStream ps = new PrintStream(new File("List.out"));
ps.print(x) // ghi 1 từ
```

43

Sử dụng Kiểu danh sách liên kết trong java

```
import java.io. *;           //import java.io.File; import java.io.FileNotFoundException; import java.io.PrintStream;
import java.util.*;
public class List1 {
    LinkedList <String> L;
    List1(){ L = new LinkedList<String>(); }
    void readFormFile() throws FileNotFoundException{
        Scanner kb = new Scanner(new File("List.inp"));
        while(kb.hasNextLine()){
            try{
                String temp = kb.next(); if(!L.contains(temp)) L.addLast(temp);
            } catch(Exception ex) { System.out.println("Error reading file!"); }
        }
    }
}
```

44

Sử dụng Kiểu danh sách liên kết trong java

```

void writeToFile() throws FileNotFoundException{
    PrintStream ps = new PrintStream(new File("List.out"));
    for(int i=0;i<L.size();i++) ps.print(L.get(i)+" ");
}

public static void main(String[] args) throws FileNotFoundException {
    List1 l = new List1();
    l.readFromFile();
    l.writeToFile();
}
}

```

45

Sử dụng Kiểu danh sách liên kết trong java

Bài tập 2:

Cho file LIST2.inp chứa các từ. Dùng kiểu L của Java viết chương trình:

- Đọc các từ từ file dữ liệu sau đó cho vào danh sách liên kết
- Ghi các từ trong danh sách ra file LIST2.out theo dạng:
 - + Từ1 số lần xuất hiện
 - + Từ2 số lần xuất hiện

Yêu cầu các từ trong file theo thứ tự từ điển, không phân biệt chữ thường chữ hoa.

Tên chương trình: List2.java

46

Tài liệu đọc thêm về danh sách liên kết

<https://www.geeksforgeeks.org/data-structures/linked-list/>

<https://www.hackerearth.com/practice/data-structures/linked-list/singly-linked-list/tutorial/>

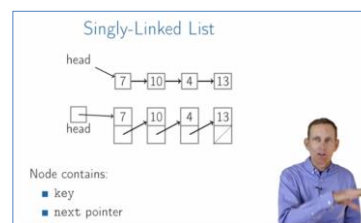
<https://www.programiz.com/dsa/linked-list-types>

47

Link YouTube

<https://www.youtube.com/watch?v=NgdwfP7K5n8>

<https://es.coursera.org/lecture/data-structures/singly-linked-lists-kHhgK>



48