



CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

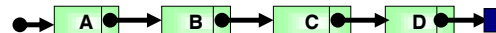
Data Structures & Algorithms

DANH SÁCH LIÊN KẾT KÉP



Các loại danh sách liên kết

- **Danh sách liên kết đơn:** Mỗi phần tử liên kết với phần tử đứng sau nó trong danh sách



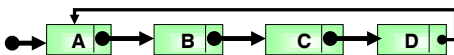
- **Danh sách liên kết kép:** Mỗi phần tử liên kết với phần tử đứng trước và sau nó trong danh sách



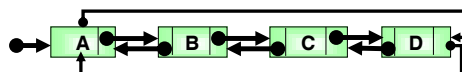
- **Danh sách liên Vòng:** Phần tử cuối danh sách liên với phần tử đầu danh sách

- **Danh sách liên Vòng:** Phần tử cuối danh sách liên với phần tử đầu danh sách

- Danh sách liên kết đơn vòng



- Danh sách liên kết đôi vòng



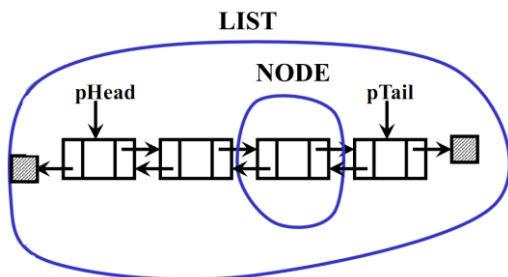
Danh sách liên kết kép

Hình ảnh:



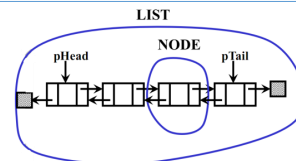
Danh sách liên kết kép

Hình ảnh:



Khai báo NODE trong DSLK Kép

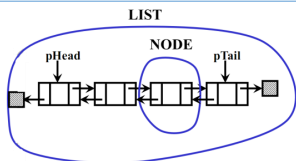
Hình ảnh:



```
11.struct node
12.{
13.|   KDL info;
14.|   struct node*pNext;
15.|   struct node*pPrev;
16.};
17.typedef struct node NODE;
```

Khai báo danh sách liên kết kép

Hình ảnh:



```
18.struct list
19.{
20.|    NODE*pHead;
21.|    NODE*pTail;
22.};
23.typedef struct list LIST;
```

Danh sách liên kết kép – Ví dụ

Ví dụ 1: Khai báo ctdl **danh sách liên kết kép** lưu trữ các số nguyên

```
10.struct node
11.{
12.|    int info;
13.|    struct node*pNext;
14.|    struct node*pPrev;
15.};
16.typedef struct node NODE;
17.struct list
18.{
19.|    NODE*pHead;
20.|    NODE*pTail;
21.};
22.typedef struct list LIST;
```

Danh sách liên kết kép – Ví dụ

Ví dụ 2: Khai báo CTDL danh sách liên kết kép lưu trữ tọa độ các điểm trong mặt phẳng oxy

```
11.struct diem
12.{
13.|    float x;
14.|    float y;
15.};
16.typedef struct diem DIEM;
```

Danh sách liên kết kép – Ví dụ

Ví dụ 2: Khai báo CTDL danh sách liên kết kép lưu trữ tọa độ các điểm trong mặt phẳng oxy

```
17.struct node
18.{
19.|    DIEM info;
20.|    struct node*pNext;
21.|    struct node*pPrev;
22.};
23.typedef struct node NODE;
24.struct list
25.{
26.|    NODE*pHead;
27.|    NODE*pTail;
28.};
29.typedef struct list LIST;
```

Khởi tạo DSLK kép

Khái niệm: là **tạo ra danh sách rỗng** không chứa node nào hết

```
1. void Init(LIST &l)
2. {
3. |    l.pHead = NULL;
4. |    l.pTail = NULL;
5. }
```

Kiểm tra DSLK kép xem có rỗng không

```
1. int IsEmpty(LIST l)
2. {
3. |    if (l.pHead==NULL)
4. |        return 1;
5. |    return 0;
6. }
```

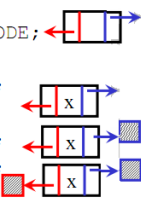
Tạo NODE cho DSLK kép

Khái niệm: là quá trình xin cấp phát vùng nhớ có kích thước bằng với kích thước của dữ liệu NODE để chứa thông tin đã được biết trước.

```

10.NODE* GetNode(KDL x)
11.{
12.    NODE *p = new NODE;
13.    if(p==NULL)
14.        return NULL;
15.    p->info = x;
16.    p->pNext = NULL;
17.    p->pPrev = NULL;
18.    return p;
19.}

```



Tạo NODE cho DSLK kép – Ví dụ

Ví dụ 1: Định nghĩa hàm tạo một NODE DSLK kép các số nguyên.

```

11.NODE* GetNode(int x)
12.{
13.    NODE *p = new NODE;
14.    if(p==NULL)
15.        return NULL;
16.    p->info = x;
17.    p->pNext = NULL;
18.    p->pPrev = NULL;
19.    return p;
20.}

```

Tạo NODE cho DSLK kép – Ví dụ

Ví dụ 2: Định nghĩa hàm tạo một NODE DSLK kép các điểm trong hệ trục tọa độ oxy.

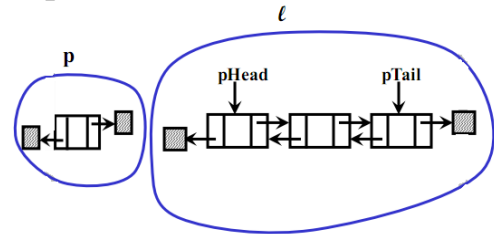
```

11.NODE* GetNode(DIEM P)
12.{
13.    NODE *p = new NODE;
14.    if(p==NULL)
15.        return NULL;
16.    p->info = P;
17.    p->pNext = NULL;
18.    p->pPrev = NULL;
19.    return p;
20.}

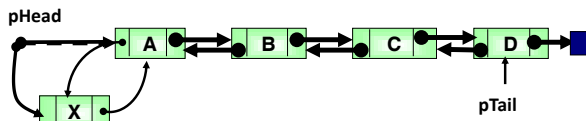
```

Thêm một NODE vào đầu DSLK kép

Hình ảnh:

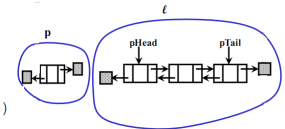


Thêm một NODE vào đầu DSLK kép



Thêm một NODE vào đầu DSLK kép

Hình ảnh:



```

11.void AddHead(LIST& l, NODE*p)
12.{
13.    if (l.pHead==NULL)
14.        l.pHead = l.pTail = p;
15.    else
16.    {
17.        p->pNext = l.pHead;
18.        l.pHead->pPrev = p;
19.        l.pHead = p;
20.    }
21.}

```

Nhập DSLK kép từ bàn phím bằng AddHead

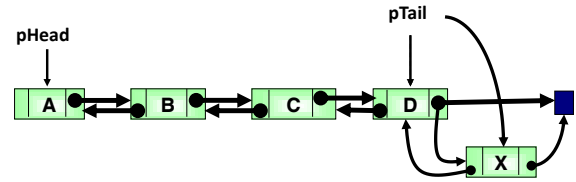
```

11. void Input (LIST &l)
12. {
13.     int n;
14.     printf("Nhap n: ");
15.     scanf("%d", &n);
16.     Init(l);
17.     for (int i=1; i<=n; i++)
18.     {
19.         KDL x;
20.         Nhap(x);
21.         NODE*p = GetNode(x);
22.         if (p!=NULL)
23.             AddHead(l, p);
24.     }
25. }

```

19

Thêm một NODE vào cuối DSLK Kép



20

Thêm một NODE vào cuối DSLK Kép

```

1. void AddTail (LIST &l, NODE*p)
2. {
3.     if (l.pHead==NULL)
4.         l.pHead=l.pTail=p;
5.     else
6.     {
7.         l.pTail->pNext=p;
8.         p->pPrev = l.pTail;
9.         l.pTail = p;
10.    }
11. }

```

21

Thêm một NODE vào cuối DSLK Kép

```

1. void AddTail (NODEPTR &ptr,
2.                NODE*p)
3. {
4.     if (ptr==NULL)
5.     {
6.         ptr = p;
7.         return;
8.     }
9.     NODE*last=ptr;
10.    while (last->pNext!=NULL)
11.        last = last->pNext;
12.    last->pNext=p;
13.    p->pPrev = last;

```

22

Thêm một NODE vào cuối DSLK Kép

```

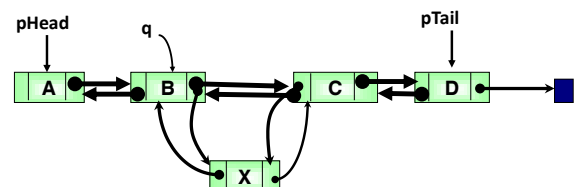
1. void AddTail (NODEPTR &ptr,
2.                NODE*p)
3. {
4.     if (ptr==NULL)
5.     {
6.         ptr = p;
7.         return;
8.     }
9.     NODE*last=ptr;
10.    while (last->pNext!=NULL)
11.        last = last->pNext;
12.    last->pNext=p;
13.    p->pPrev = last;

```

23

Thêm một NODE vào sau NODE Q trong DSLK Kép

- Minh họa thêm nút X vào sau nút q



24

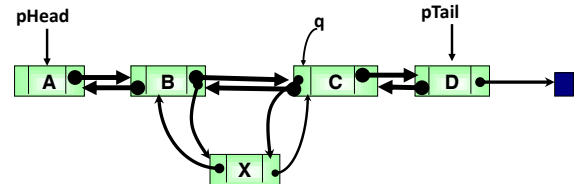
Thêm một NODE vào sau NODE Q trong DSLK Kép

```
void AddLastQ(DList &l, NODE *x, NODE *q)
{
    NODE *p;
    p = q->pNext;
    if (q != NULL) // them vao duoc
    {
        x->pNext = p;
        x->pPre = q;
        q->pNext = x;
        if (p != NULL)
            p->pPre = x;
        if (q == l.pTail) // them vao sau danh sach lien ket.
            l.pTail = x;
    }
    else
        AddHead(l, x);
}
```

25

Thêm một NODE vào trước NODE Q trong DSLK Kép

- Minh họa thêm nút X vào trước nút q



Thêm một NODE vào trước NODE Q trong DSLK Kép

```
void AddBeforeQ(List &l, NODE *x, NODE *q)
{
    Node *p;
    p = q->pPre;
    if (q != NULL)
    {
        x->pNext = q;
        q->pPre = x;
        x->pPre = p;
        if (p != NULL)
            p->pNext = x;
        if (q == l.pHead)
            l.pHead = x;
    }
    else
        AddTail(l, tam);
}
```

Duyệt DSLK Kép

Khái niệm: Duyệt tuần tự là **duyet qua tất cả các node trong DSLK**

```
11. KDL <Tên Hàm> (LIST l)
12. {
13.     ...
14.     NODE *p = l.pHead;
15.     while (p != NULL)
16.     {
17.         ...
18.         p = p->pNext;
19.     }
20.     ...
21. }
```

26

Duyệt DSLK Kép

Khái niệm: Duyệt tuần tự là **duyet qua tất cả các node trong DSLK**

```
11. KDL <Tên Hàm> (LIST l)
12. {
13.     ...
14.     NODE *p = l.pTail;
15.     while (p != NULL)
16.     {
17.         ...
18.         p = p->pPrev;
19.     }
20.     ...
21. }
```

27

Duyệt DSLK Kép – Ví dụ

Ví dụ 1: Định nghĩa hàm xuất DSLK kép các số nguyên

```
11. void Xuat (LIST l)
12. {
13.     NODE *p = l.pHead;
14.     while (p != NULL)
15.     {
16.         printf("%4d", p->info);
17.         p = p->pNext;
18.     }
19. }
```

28

Duyệt DSLK Kép – Ví dụ

Ví dụ 2: Định nghĩa hàm **tính tổng các số lẻ** trong DSLK kép các số nguyên

```

11.int TongLe (LIST l)
12.{
13.    int s = 0;
14.    NODE*p = l.pHead;
15.    while (p!=NULL)
16.    {
17.        if (p->info%2!=0)
18.            s = s + p->info;
19.        p = p->pNext;
20.    }
21.    return s;
22.}

```

31

Duyệt DSLK Kép – Ví dụ

Ví dụ 2: Định nghĩa hàm **tính tổng các số lẻ** trong DSLK kép các số nguyên

```

11.int TongLe (LIST l)
12.{
13.    int s = 0;
14.    NODE*p = l.pTail;
15.    while (p!=NULL)
16.    {
17.        if (p->info%2!=0)
18.            s = s + p->info;
19.        p = p->pPrev;
20.    }
21.    return s;
22.}

```

32

Lấy NODE đầu của DSLK Kép

```

1. NODE* GetHead (LIST &l)
2. {
3.     if (l.pHead==NULL)
4.         return NULL;
5.     if (l.pHead==l.pTail)
6.     {
7.         NODE*p=l.pHead;
8.         l.pHead=l.pTail=NULL;
9.         return p;
10.    }
11.    NODE*p = l.pHead;
12.    l.pHead = l.pHead->pNext;
13.    l.pHead->pPrev = NULL;
14.    p->pNext = NULL;
15.    return p;
16.}

```

33

Lấy NODE đầu của DSLK Kép

```

1. NODE* GetHead (NODEPTR &ptr)
2. {
3.     if (ptr==NULL)
4.         return NULL;
5.     if (ptr->pNext==NULL)
6.     {
7.         NODE*p = ptr;
8.         ptr = NULL;
9.         return p;
10.    }
11.    NODE*p = ptr;
12.    ptr = ptr->pNext;
13.    ptr->pPrev = NULL;
14.    p->pNext = NULL;
15.    return p;
16.}

```

34

Lấy NODE cuối của DSLK Kép

```

1. NODE* gettail (LIST &l)
2. {
3.     if (l.pHead==NULL)
4.         return NULL;
5.     if (l.pHead==l.pTail)
6.     {
7.         NODE*p=l.pHead;
8.         l.pHead=l.pTail=NULL;
9.     }
10.    NODE*p=l.pTail;
11.    NODE*q=l.pHead;
12.    while (q->pNext!=l.pTail)
13.        q=q->pNext;
14.    l.pTail=q;
15.    l.pTail->pNext=NULL;
16.    return p;
17.}

```

35

Xóa NODE đầu của DSLK Kép

```

void DeleteFirst (List &l)
{
    NODE *p;
    if (l.pHead!=NULL)
    {
        p=l.pHead;
        l.pHead=l.pHead->pNext;
        l.pHead->pPre=NULL;
        delete p;
        if (l.pHead==NULL)
            l.pTail=NULL;
    }
}

```

Xóa NODE cuối của DSLK Kép

```
void DeleteEnd(List &l)
{
    NODE *p;
    if(l.pHead!=NULL)
    {
        p=l.pTail;
        l.pTail=l.pTail->Pre;
        l.pTail->pNext=NULL;
        delete p;
        if(l.pTail==NULL)
            l.pHead=NULL;
    }
}
```

Xóa NODE sau NODE Q trong DSLK Kép

```
void DeleteLastQ(List &l, NODE *q)
{
    NODE *p; //luu node dung sau node q
    if(q!=NULL)
    {
        p=q->pNext;
        if(p!=NULL)
        {
            q->pNext=p->pNext;
            if(p==l.pTail) //xoa dung nut cuoi
                l.pTail=q;
            else //Nut xoa khong phai nut cuoi
                p->pNext->pPre=q;
            delete p;
        }
    }
    else
        DeleteFirst(l);
}
```

Xóa NODE trước NODE Q trong DSLK Kép

```
void DeleteBeforeQ(DList &l, DNode *q)
{
    DNode *p;
    if(q!=NULL) //tuc ton tai node q
    {
        p=q->pPre;
        if(p!=NULL)
        {
            q->pPre=p->pPre;
            if(p==l.pHead) //p la Node dau cua danh sach
                l.pHead=q;
            else //p khong phai la node dau
                p->pPre->pNext=q;
            delete p;
        }
    }
    else
        DeleteEnd(l);
}
```

Xóa NODE có thông tin X trong DSLK Kép

```
int DeleteX(List &l, int x)
{
    Node *p;
    Node *q;
    q=NULL;
    p=l.pHead;
    while(p!=NULL)
    {
        if(p->Info==x)
            break;
        q=p; //q la Node co truong Info = x
        p=p->pNext;
    }
    if(q==NULL) return 0; //khong tim thay Node nao co truong Info =x
    if(q!=NULL)
        DeleteLastQ(l,q);
    else
        DeleteFirst(l);
    return 1;
}
```

Sắp xếp DSLK Kép

```
void DoiChoTrucTiep(List &l)
{
    NODE *p,*q;
    p=l.pHead;
    while(p!=l.pTail)
    {
        q=p->pNext;
        while(q!=NULL)
        {
            if(p->Info>q->Info)
                swap(p,q);
            q=q->pNext;
        }
        p=p->pNext;
    }
}
```

Bài Tập

Bài toán: Viết chương trình thực hiện các yêu cầu sau:

- + Nhập dslk kép các phân số.
- + Đếm số lượng giá trị dương trong dslk kép.
- + Xuất dslk kép.

Slide được tham khảo từ

- **Slide được tham khảo từ:**

- Slide CTDL GT, Khoa Khoa Học Máy Tính, ĐHCNTT
- Slide CTDL GT, Thầy Nguyễn Tấn Trần Minh Khang, ĐH CNTT
- Congdongcviet.com
- Cplusplus.com



43

44