

HK I, 2012-2013

# Bài 4: Cấu trúc dữ liệu biểu diễn danh sách (P2)

Giảng viên: Hoàng Thị Điệp

Khoa Công nghệ Thông tin – Đại học Công Nghệ

# Kiểm tra chuẩn bị bài (giấy, 20 phút)

- Đề nghị ghi rõ họ tên, mã sinh viên, nhóm thực hành, tên tài khoản bitbucket vào giấy.
  - Được dùng tài liệu! Không được trao đổi!
1. Bộ công cụ lập là gì? Theo bạn vì sao nên dùng bộ công cụ lập?
  2. Ở phần cài đặt danh sách bằng mảng động, giáo trình đề xuất 1 cách xử lý mảng đầy khi cần chèn thêm phần tử mới (hàm insert hoặc append). Hãy trình bày cách xử lý này và bàn luận về ưu nhược điểm của nó.
  3. (CLC) Trong C++, tham chiếu là gì? Cho ví dụ.

# Giới thiệu thư viện khuôn mẫu chuẩn STL

- `<vector>`
- `<deque>`
- `<list>`
- `<stack>`
- `<queue>`
- `<priority_queue>`
- `<set>`
- `<multiset>`
- `<map>`
- `<multimap>`
- `<bitset>`

# Bài thực hành số 3

```
// Lớp ArrayList biểu diễn danh sách số nguyên cài bằng mảng tĩnh
class ArrayList{
public:
    static const int MAX_SIZE = 100; // Số phần tử tối đa của mảng
    ArrayList(); // Khởi tạo danh sách rỗng

    bool empty() const; // Kiểm tra DS rỗng hay không
    int length() const; // Xác định độ dài DS
    void insert(const int x, int i); // Xen giá trị x vào vị trí i trong DS
    void append(const int x); // Thêm giá trị x vào đuôi DS
    void erase(int i); // Loại khỏi DS phần tử ở vị trí i
    int& at(int i) const; // Trả về tham chiếu đến phần tử ở vị trí i
    void print() const; // In ra màn hình toàn bộ DS
private:
    int element[MAX_SIZE]; // Mảng tĩnh element lưu các phần tử của DS
    int current; // chỉ số của phần tử hiện tại
    int last; // chỉ số của phần tử cuối cùng
};
```

# Ôn tập

- Con trỏ và bộ nhớ động
- Bộ công cụ lập

# Cài đặt danh sách bằng mảng C++

## Cấp phát tĩnh

```
template <class Item>
class List{
public:
    static const int MAX = 50;
    // ...
private:
    Item element[MAX];
    int last;
};
```

## Cấp phát động

```
template <class Item>
class Dlist{
public:
    // ...
private:
    Item * element;
    int size;
    int last;
};
```

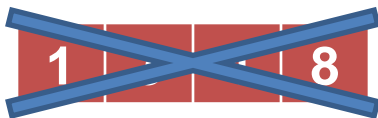
# Bộ ba quan trọng

Dlist có thành phần dữ liệu cấp phát động nên phải cài đặt bộ ba

- Hàm kiến tạo sao chép
- Toán tử gán
- Hàm hủy

# Hàm insert, append của Dlist

- $A = (1, 3, 4, 8)$ ; size = 4; last = 3
- insert( $A, 3, 50$ )



$A = (1, 3, 4, 50, 8)$ ; size = 8; last = 4



# Hàm insert, append của Dlist

Khi mảng đầy

- Cấp phát động một mảng mới có cỡ gấp đôi mảng cũ
- Chép đoạn đầu của mảng cũ sang mảng mới
- Đưa phần tử cần xen vào mảng mới
- Chép đoạn còn lại của mảng cũ sang mảng mới
- Hủy mảng cũ
- Cập nhật size, last

Viết mã C++!

# Ứng dụng KDLT danh sách

- **Tập động**

- Mỗi phần tử có thành phần khóa phân biệt. Các giá trị khóa có quan hệ thứ tự
- Các phép toán: isEmpty, insert, del, search, getMax, getMin
- Ví dụ: ((“An”, 1985), (“Bình”, 1986), (“Cường”, 1985), (“Dung”, 1987))
- Cài bằng danh sách được sắp hay không được sắp thì tốt hơn?

# KDLTT tập động

```
1 class IntSet{
2 public:
3     IntSet();
4     IntSet(int _size);
5     IntSet(const IntSet& _is);
6     ~IntSet();
7
8     bool isEmpty() const;
9     bool search(int x) const;
10    int getMax() const;
11    int getMin() const;
12
13    void insert(int x);
14    void del(int x);
15
16    friend ostream& operator<<(ostream& _out, const IntSet& _is);
17    friend istream& operator>>(istream& _in, IntSet& _is);
18
19    IntSet& operator=(const IntSet& _is);
20    const IntSet operator|(const IntSet& _is);
21    const IntSet operator&(const IntSet& _is);
22    const IntSet operator-(const IntSet& _is);
23    bool operator<=(const IntSet& _is);
24
25 private:
26     int * element;
27     int size;
28     int last;
29 };
```

# Ứng dụng KDLTT danh sách

- **Đa thức**

- Ví dụ:  $((17,5), (-25, 2), (14, 1), (-32, 0))$  biểu diễn đa thức  $17x^5 - 25x^2 + 14x - 32$
- Các phép toán: cộng, trừ, nhân

# Ứng dụng KDUTT danh sách

- **Ma trận thưa**

- Ma trận chỉ chứa một số ít các phần tử khác 0
- Cách biểu diễn:
  - Xem ma trận như danh sách các dòng
  - Mỗi dòng là một danh sách biểu diễn các phần tử khác 0
  - Mỗi phần tử khác 0 là một cặp (chỉ số cột, giá trị)
- Ví dụ: (((2, 7), (5, 3)), ((3, 8)), ()), ((2, 5), (4, 9)))
- Các phép toán trên ma trận, trên dòng, trên phần tử

# Tìm kiếm nhị phân

Algorithm *binarySearch*(*x*, *A*, *first*, *last*):

**Input:** search keyword *x*

and array *A* of Items with two ends marked by indexes *first* and *last*

**Output:** **true** if *x* in *A*, **false** otherwise

**if** *first* > *last* **then**

**return** false

*mid*  $\leftarrow$  (*first* + *last*) / 2

**if** *x* = *A*[*mid*].key **then**

**return** true

**else if** *x* < *A*[*mid*].key **then**

*binarySearch*(*x*, *A*, *first*, *mid* - 1)

**else**

*binarySearch*(*x*, *A*, *mid* + 1, *last*)

# Tìm kiếm nhị phân

A →	1	3	4	6	8	9	11
chỉ số →	0	1	2	3	4	5	6

$A = (1, 3, 4, 6, 8, 9, 11); x = 4.$   
 $\text{search}(x, A).$

- `binarySearch(x, A, first, last)`
- `binarySearch(x, A, 0, 6)`
  - So  $x$  với  $A[3]$ .  $x$  nhỏ hơn.
- `binarySearch(x, A, 0, 2)`
  - So  $x$  với  $A[1]$ .  $x$  lớn hơn.
- `binarySearch(x, A, 2, 2)`
  - So  $x$  với  $A[2]$ . Bằng. Trả về true.

# Chuẩn bị bài tới

- Đọc chương 5 giáo trình.