C Programming Basic – week 6

Nhu cầu tìm kiếm

- Hoạt động hàng ngày – trang vàng, trường đại học, hiệu làm đầu
- Máy tính có thể thực hiện tác vụ tìm kiếm
- World wide web – các cơ chế tìm kiếm khác nhau, yahoo.com, ask.co.uk, google.com
- Spreadsheet –danh sách các tên
- Databases – tìm kiếm bản ghi - *select* * *from* ...
- Số lượng bản ghi lớn –tốn thời gian - số lượng phép so sánh lớn

Nội dung

1. Tìm kiếm tuần tự
2. Tìm kiếm tự tổ chức
3. Tìm kiếm nhị phân

1. Tìm kiếm tuần tự

- Tìm kiếm tuyến tính
- Thăm tất cả các phần tử từ phần tử đầu tiên
- So sánh khóa với từng phần tử
- Nếu tìm thấy, trả về chỉ số ; nếu không trả về -1
- Các phần tử của danh sách có trật tự bất kỳ

Thuật toán

```
int LinearSearch(T M[], int N, T
  X){
  int k = 0;
  while (M[k] != X && k < N) k++;
  if (k < N) return k;
  return -1;
}
```

VD

```
#include<stdio.h>

 int sequential_search(char *items, int count, char key)
  {
    register int t;

    for(t=0; t < count; ++t)
      if(key == items[t]) return t;
    return -1; /* no match */
  }

  int main(void){
     char *str = "asdf";

     int index = sequential_search(str, 4, 's');

     printf("%d",index);

  }
```

Phần tử sentinel

- Trong tìm kiếm tuần tự, mỗi phép so sánh cần các thao tác
  - Kiểm tra hai điều kiện
  - Thực thi một statement
- Có thể tránh kiểm tra điều kiện kết thúc danh sách bằng cách thêm phần tử sentinel vào cuối danh sách

Thuật toán

- Tìm kiếm tuần tự từ vị trí bắt đầu cho tới khi tìm thấy (chắc chắn tìm thấy)
- Nếu trả về vị trí cuối cùng, nghĩa là tìm thấy phần tử sentinel, thuật toán tìm kiếm thất bại
- Nếu không chỉ số trả về là của phần tử đầu tiên tìm thấy trong danh sách

Thuật toán

```
int LinearSentinelSearch(T M[],
  int N, T X){
  int k = 0; M[N]=X;
  while (M[k] != X)
  k++;
  return k-1;
}
```

Exercise 6.1

- Đọc 11 số nguyên từ bàn phím và lưu 10 số đầu tiên vào mảng
- Nếu số thứ 11 có trong mảng, in ra vị trí, nếu không, in ra -1

Exercise 6.2

- Xây dựng danh bạ điện thoại
- Khai báo một cấu trúc "Address" chứa ít nhất name, telephone number và e-mail address. Chương trình có thể xử lý danh sách chứa 100 phần tử
- Đọc 10 địa chỉ từ tệp và lưu vào danh sách; tìm kiếm theo tên bằng tìm kiếm tuần tự và viết địa chỉ tìm thấy đầu tiên ra màn hình
  (1) Sử dụng mảng
  (2) Sử dụng danh sách liên kết đơn hoặc đôi

Exercise 6.3

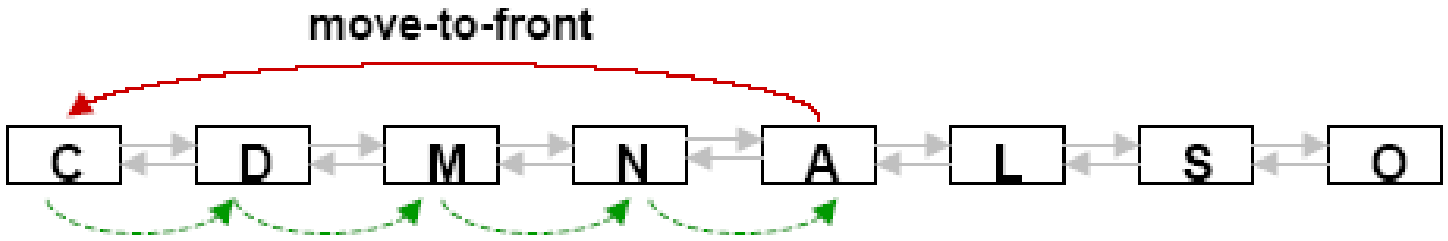
- Cho hai danh sách có cùng kiểu, tìm :

(a) các bản ghi có trong list1 nhưng không có trong list2

(b) các bản ghi có trong list2 nhưng không có trong list1

2. Tìm kiếm tự tổ chức

- Move-to-front
- Transpose



Move-to-front

```
int search(int key,int r[], int n){
   int i,j;
   int temp;
   for (i = 0; i < n-1 && r[i] != key; i++);
   if (key == r[i]){
    if (i > 0){
          temp = r[i];
          for (j = i-1, j >= 0; j--) r[j+1] = r[j];
          r[0] = temp;
       }
       return i;
   }else return -1;
}
```

Transpose

```
int search(int key, int r[], int n ) {
   int i;
   int temp;
   for (i = 0; i < n && r[i] != key; i++);
   if (key == r[i]) {
    if (i > 0) {
          /*** Transpose with predecessor ***/
          temp = r[i];
          r[i] = r[i-1];
          r[--i] = temp;
      };
      return i;
   }else return -1;
}
```

Exercise 6.4

- Sửa đổi hàm tìm kiếm trong Exercise 6.1 thành tìm kiếm tự tổ chức sử dụng:
  (1) move-to-front
  (2) transpose

Homework 1

- Cài đặt move-to-front and transpose cho thư viện List
- Sử dụng trong bài Nokia DB :
  - Tìm kiếm và cập nhật theo tên model
  - Sử dụng menu từ các bài tập trước

Homework 2

- Một từ điển được lưu trữ tại /usr/share/dict/words
- Viết chương trình nhận vào một chuỗi và in ra tất cả các từ bắt đầu bằng chuỗi đó

[Example]

```
% look computer
computer
computerize
computerized
computerizes
computerizing
computers
```