ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LẬP TRÌNH C CƠ BẢN

Sắp xếp – phần 2

NỘI DUNG

- Sinh dữ liệu thử nghiệm
- Cài đặt 3 thuật toán sắp xếp trộn, vun đống, sắp xếp nhanh
- Thí nghiệm và đánh giá

Bài tập 1

- Hồ sơ của mỗi nhân viên bao gồm các trường thông tin sau
  - Họ và tên
  - Ngày, tháng, năm sinh
- Viết chương trình sinh ra n hồ sơ một cách ngẫu nhiên và ghi ra file profile-n.txt với định dạng
  - Dòng 2i-1 và dòng 2i (i=1,…,n) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
  - Dòng 2n+1: ghi ký tự # thể hiện dấu hiệu kết thúc file

Profile-5.txt

```
Bui Hai An
1980-02-30
Pham Viet Anh
1986-10-08
Do Duc Bang
1990-04-24
Dang Van Cuong
1987-08-17
Pham Viet Anh
1986-05-20
#
```

SẮP XẾP TRỘN

- Dựa trên chia để trị
- Chia dãy a_1, …, a_n thành 2 dãy con có độ dài bằng nhau
- Sắp xếp 2 dãy con bằng thuật toán sắp xếp trộn
- Trộn 2 dãy con đã được sắp với nhau để thu được dãy ban đầu được sắp thứ tự

```
void mergeSort(int A[], int L, int R) {
  if(L < R){
    int M = (L+R)/2;
    mergeSort(A,L,M);
    mergeSort(A,M+1,R);
    merge(A,L,M,R);
  }
}
```

SẮP XẾP TRỘN

- Sử dụng mảng trung gian để lưu trữ tạm thời trong quá trình trộn

```
void merge(int A[], int L, int M, int R) {
// tron 2 day da sap A[L..M] va A[M+1..R]
  int i = L; int j = M+1;
  for(int k = L; k <= R; k++){
    if(i > M){ TA[k] = A[j]; j++;}
    else if(j > R){TA[k] = A[i]; i++;}
    else{
      if(A[i] < A[j]){
         TA[k] = A[i]; i++;
      }
      else {
         TA[k] = A[j]; j++;
      }
    }
  }
  for(int k = L; k <= R; k++) A[k] = TA[k];
}
```

Bài tập 2

- Viết chương trình nhập vào dãy n hồ sơ từ các file dữ liệu được sinh ra trong Bài tập 1 (ở trên), thực hiện sắp xếp các hồ sơ theo thứ tự không giảm (ưu tiên so sánh họ tên trước, rồi mới đến ngày, tháng, năm sinh) bằng thuật toán sắp xếp trộn
- Dữ liệu (profile-n.txt)
  - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
  - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file
- Kết quả (sorted-profile-n.txt)
  - Ghi dãy đã được sắp xếp theo thứ tự với định dạng
    - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
    - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file

Bài tập 2

- Ví dụ

Profile-5.txt	Sorted-Profile-5.txt
Dang Van Cuong 1987-08-17 Pham Viet Anh 1986-10-08 Bui Hai An 1980-02-30 Do Duc Bang 1990-04-24 Pham Viet Anh 1986-05-20 #	Bui Hai An 1980-02-30 Pham Viet Anh 1988-10-08 Pham Viet Anh 1986-05-20 Do Duc Bang 1990-04-24 Dang Van Cuong 1987-08-17 #

SẮP XẾP NHANH

- Chọn một phần tử bất kỳ dùng làm phần tử trụ (pivot)
- Sắp xếp lại dãy sao cho
  - Các phần tử đứng trước phần tử trụ sẽ không lớn hơn phần tử trụ
  - Các phần tử đứng sau phần tử trụ không nhỏ hơn phần tử trụ
- Khi đó phần tử trụ (có thể bị thay đổi vị trí) đã đứng đúng vị trí trong dãy khi được sắp thứ tự
- Tiến hành sắp xếp dãy con đứng trước và sau phần tử trụ bằng sắp xếp nhanh

SẮP XẾP NHANH

```
void quickSort(int A[], int L, int R) {
  if(L < R){
    int index = (L + R)/2;
    index = partition(A, L, R, index);
    if(L < index)
      quickSort(A, L, index-1);
    if(index < R)
      quickSort(A, index+1, R);
  }
}
```

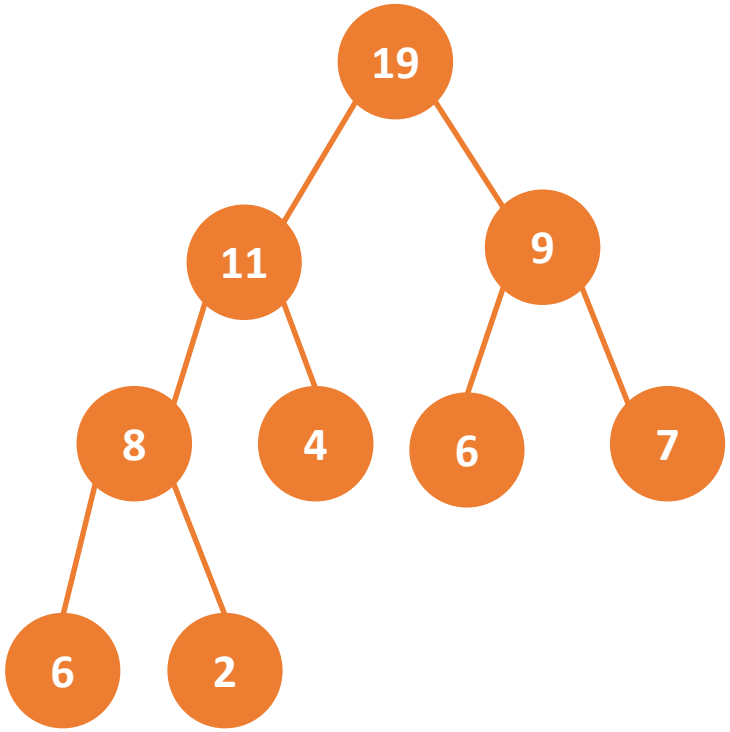
```
int partition(int A[], int L, int R, int
                          indexPivot) {
  int pivot = A[indexPivot];
  swap(A[indexPivot], A[R]);
  int storeIndex = L;
  for(int i = L; i <= R-1; i++){
    if(A[i] < pivot){
      swap(A[storeIndex], A[i]);
      storeIndex++;
    }
  }
  swap(A[storeIndex], A[R]);
  return storeIndex;
}
```

Bài tập 2

- Viết chương trình nhập vào dãy n hồ sơ từ các file dữ liệu được sinh ra trong Bài tập 1 (ở trên), thực hiện sắp xếp các hồ sơ theo thứ tự không giảm (ưu tiên so sánh họ tên trước, rồi mới đến ngày, tháng, năm sinh) bằng thuật toán sắp nhanh
- Dữ liệu (profile-n.txt)
  - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
  - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file
- Kết quả (sorted-profile-n.txt)
  - Ghi dãy đã được sắp xếp theo thứ tự với định dạng
    - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
    - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file

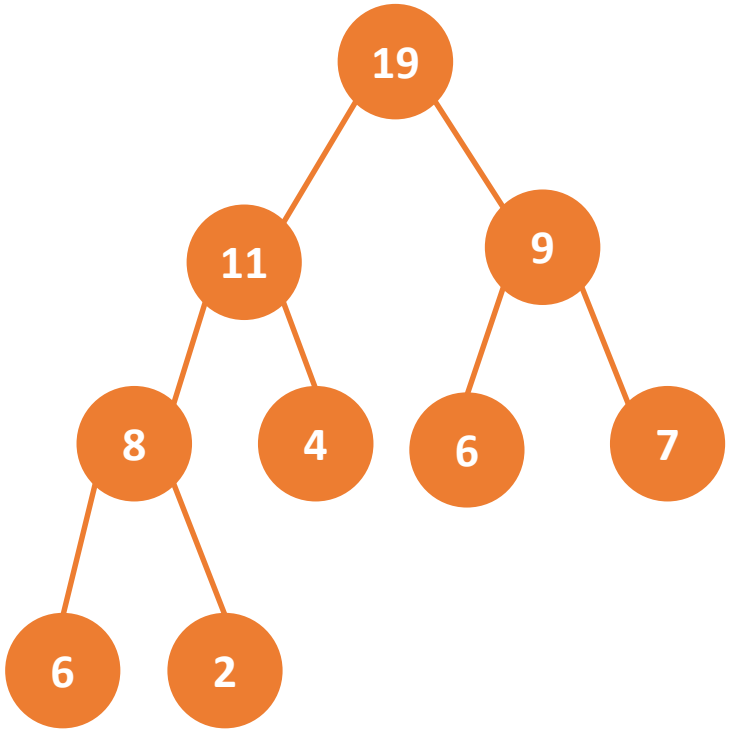
SẮP XẾP VUN ĐỐNG

- Cấu trúc đống (max-heap)
  - Cây nhị phân đầy đủ (complete tree)
  - Khoá của mỗi nút lớn hơn hoặc bằng khoá của 2 nút con (tính chất của max-heap)
- Ánh xạ từ dãy A[1…N] sang cây nhị phân đầy đủ
  - Gốc là A[1]
  - A[2i] và A[2i+1] là con trái và con phải của A[i]
  - Chiều cao của cây là logN + 1



SẮP XẾP VUN ĐỐNG

- Vun lại đống (heapify)
  - Tình trạng:
    - Tính chất max-heap ở A[i] bị phá vỡ
    - Tính chất max-heap ở các cây con của A[i] đã được thoả mãn
  - Vun lại đống để khôi phục tại tính chất max-heap trên cây gốc A[i]



SẮP XẾP VUN ĐỐNG

- Vun lại đống (heapify)
  - Chọn nút con lớn nhất
  - Đổi chỗ nút con và A[i] cho nhau nếu nút con này lớn hơn A[i] và vun lại đống bắt đầu từ nút con này

```
void heapify(int A[], int i, int N)
{
  int L = 2*i;
  int R = 2*i+1;
  int max = i;
  if(L <= N && A[L] > A[i])
    max = L;
  if(R <= N && A[R] > A[max])
    max = R;
  if(max != i){
    swap(A[i], A[max]);
    heapify(A,max,N);
  }
}
```

SẮP XẾP VUN ĐỐNG

- Sắp xếp vun đống
  - Xây dựng max-heap (thủ tục buildHeap)
  - Đổi chỗ A[1] và A[N] cho nhau
  - Vun lại đống bắt đầu từ A[1] cho A[1..N-1]
  - Đổi chỗ A[1] và A[N-1] cho nhau
  - Vun lại đống bắt đầu từ A[1] cho A[1..N-2]
  - …

```
void buildHeap(int A[], int N) {
  for(int i = N/2; i >= 1; i--)
    heapify(A,i,N);
}
void heapSort(int A[], int N) {
  // index tu 1 -> N
  buildHeap(A,N);
  for(int i = N; i > 1; i--) {
    swap(A[1], A[i]);
    heapify(A, 1, i-1);
  }
}
```

SẮP XẾP VUN ĐỐNG

- Viết chương trình nhập vào dãy n hồ sơ từ các file dữ liệu được sinh ra trong Bài tập 1 (ở trên), thực hiện sắp xếp các hồ sơ theo thứ tự không giảm (ưu tiên so sánh họ tên trước, rồi mới đến ngày, tháng, năm sinh) bằng thuật toán sắp xếp vun đống
- Dữ liệu (profile-n.txt)
  - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
  - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file
- Kết quả (sorted-profile-n.txt)
  - Ghi dãy đã được sắp xếp theo thứ tự với định dạng
    - Dòng 2i-1 và dòng 2i (i = 1,…) tương ứng ghi họ tên (không dấu) và ngày, tháng, năm sinh của hồ sơ thứ i. Họ và tên ghi theo định dạng <họ> <tên đệm> <tên> và ngày, tháng, năm sinh ghi theo định dạng YYYY-MM-DD
    - Dòng cuối cùng: ghi ký tự # thể hiện dấu hiệu kết thúc file

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!