C Programming Basic – week 10

Sorting

Lecturer :

Do Quoc Huy

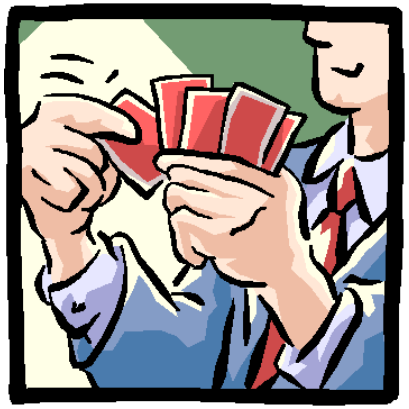
Dept of Computer Science

Hanoi University of Technology

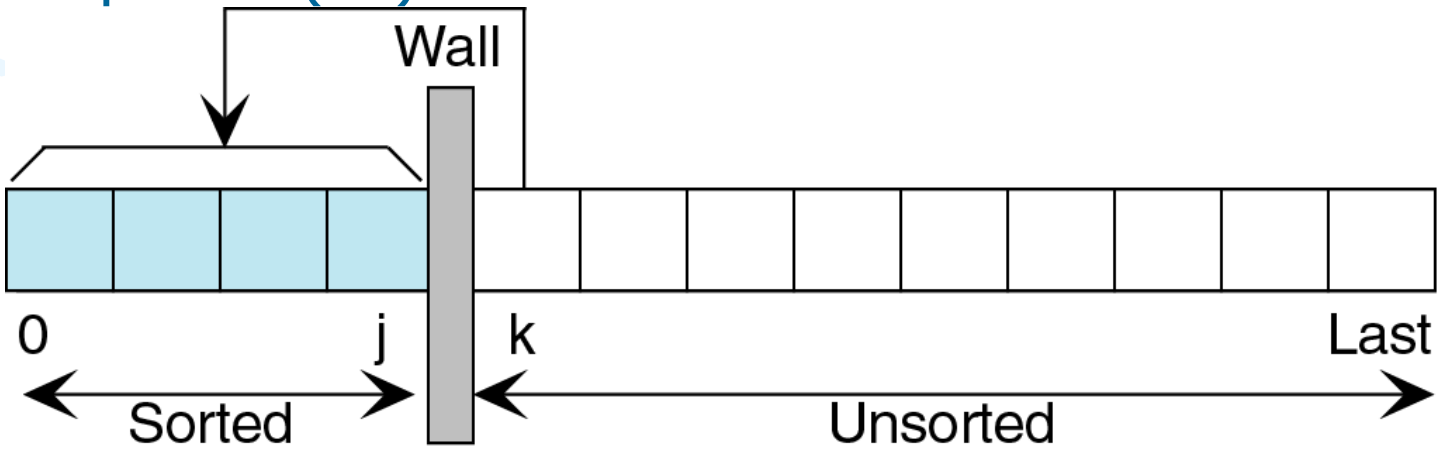
Topics of this week

- Giải thuật sắp xếp cơ bản
  - Chèn
  - Chọn
  - Nổi bọt (hoán vị)
- Giải thuật vun đống (heap sort)

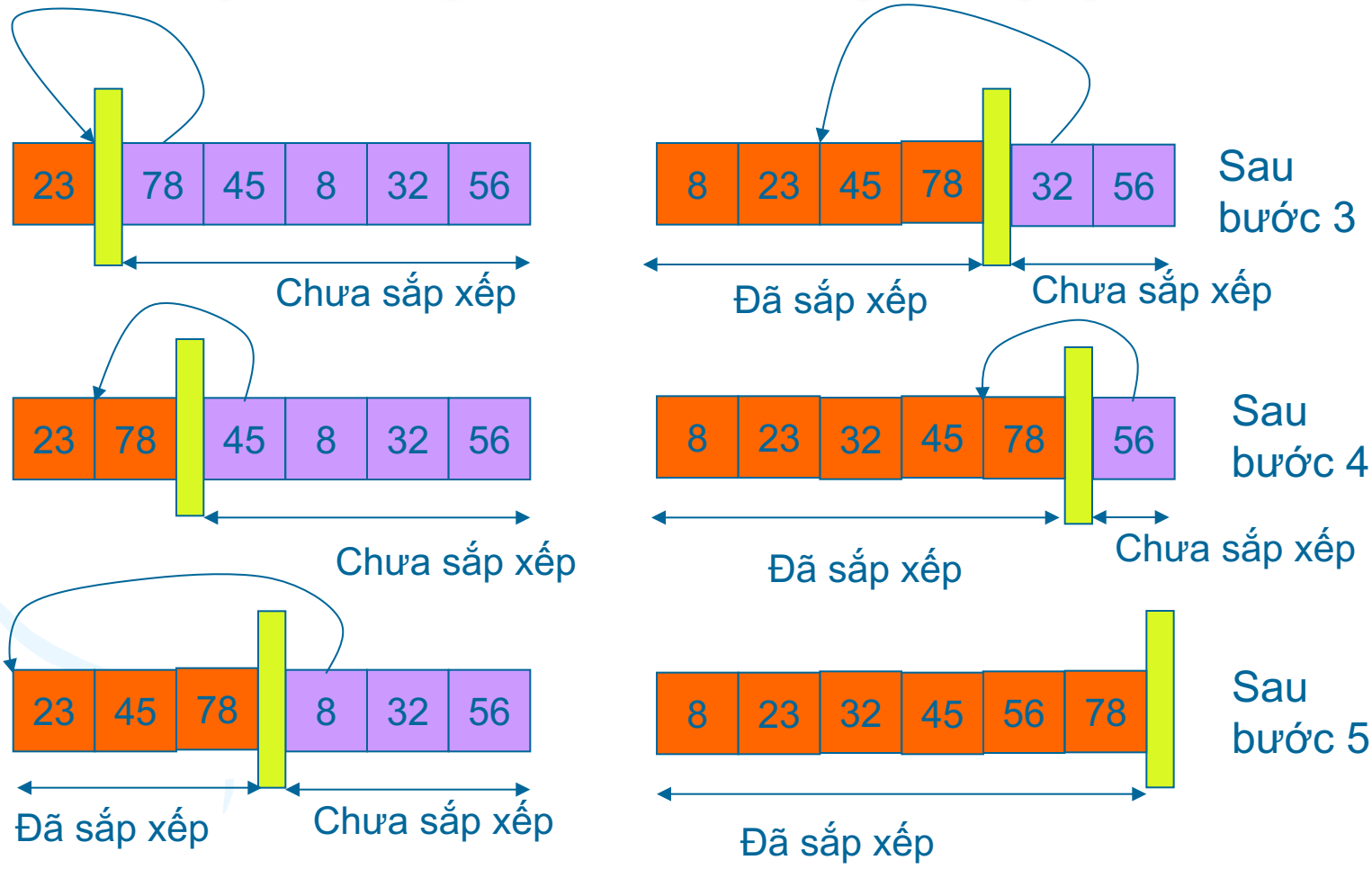
Sắp xếp chèn (Insertion)



- Chiến thuật của những người chơi bài
- Sắp xếp danh sách theo
  - Tìm phần tử đầu tiên chưa được sắp xếp trong danh sách
  - Chuyển nó tới vị trí hợp lý
  - Độ hiệu quả: O(n^2)



Sắp xếp chèn (tiếp)



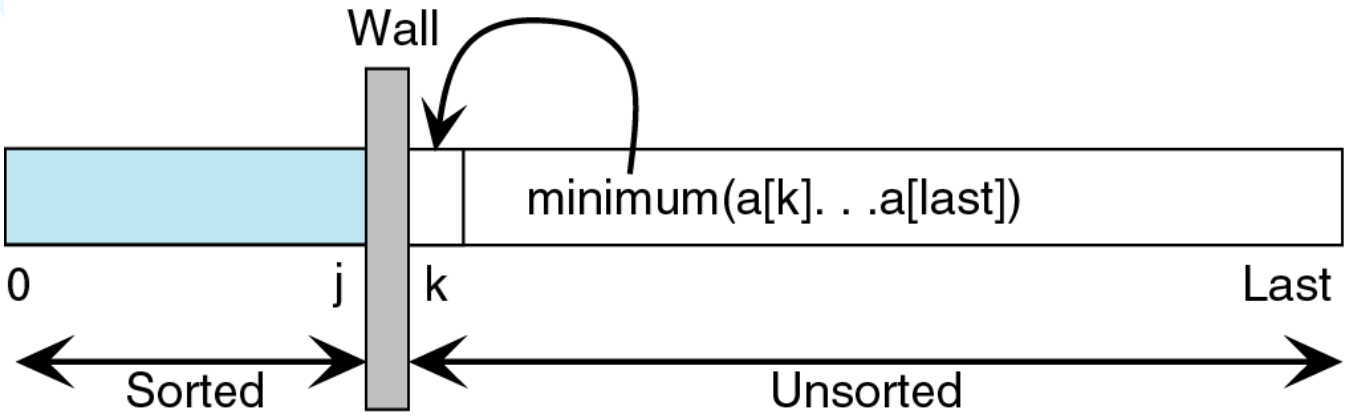
unsorted

Sắp xếp chèn (tiếp)

```
void insertion_sort(element list[], int n)
{
   int i, j;
   element next;
   for (i=1; i<n; i++) {
      next= list[i];
      for (j=i-1;j>=0 && next.key< list[j].key;
           j--)
        list[j+1] = list[j];
      list[j+1] = next;
   }
}
```

Sắp xếp chọn

- Sắp xếp danh sách theo
  - Tìm phần tử nhỏ nhất (hoặc lớn nhất) trong danh sách
  - Chuyển phần tử này lên phía đầu (hoặc cuối) danh sách bằng cách hoán đổi nó với phần tử ở vị trí đầu (hoặc cuối)



Selection sort

```
void selection(element a[], int n)
   { int i, j, min, tmp;
     for (i = 0; i < n-1; i++){
         min = i;
         for (j = i+1; j <=n-1 ; j++)
             if ( a[j].key < a[min].key)
                 min = j;
         tmp= a[i];
         a[i]= a[min]);
         a[min] = tmp;
       }
   }
```

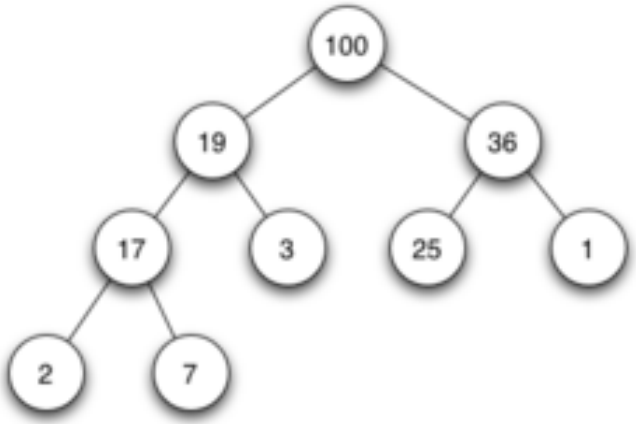
Exercise 1

- Giả sử bạn tạo một danh bạ điện thoại.
- Viết một chương trình có thể chứa ít nhất 100 danh bạ.
- Đọc khoảng 10 dữ liệu từ một file đầu vào và ghi dữ liệu đã sắp xếp theo thứ tự tăng dần ra một file đầu ra.
- Sử dụng sắp xếp chèn hoặc sắp xếp chọn

- (1) Viết chương trình sử dụng mảng các cấu trúc
- (2) Viết chương trình sử dụng danh sách liên kết.
- Trong cả 2 chương trình, in ra số phép so sánh được thực hiện trong quá trình sắp xếp của từng giải thuật.

Vun đống (Heap sort)

- Đống: Một cây nhị phân
  - Gốc được đảm bảo là giữ nút lớn nhất trên cây
  - Các giá trị nhỏ hơn sẽ ở cây con trái hoặc phải
  - Cây đầy đủ hoặc gần đầy đủ
  - Giá trị khóa ở từng nút >= giá trị khóa ở các nút con

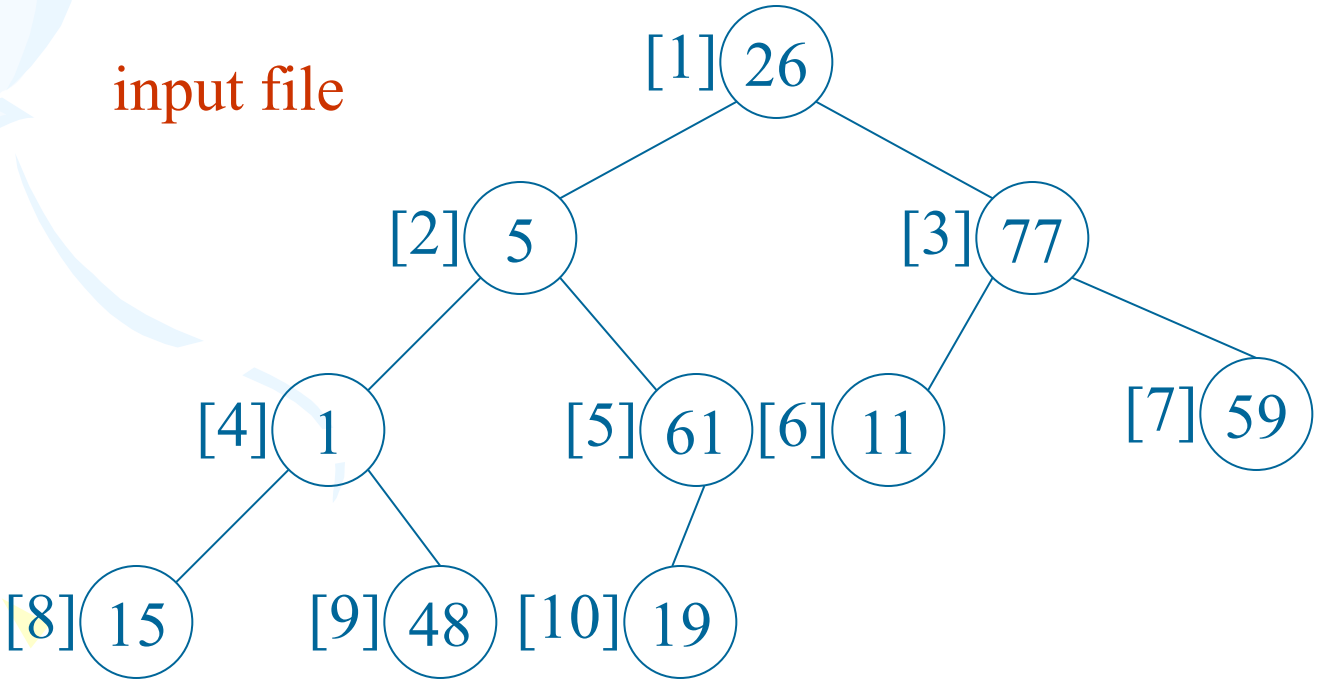


Heap sort

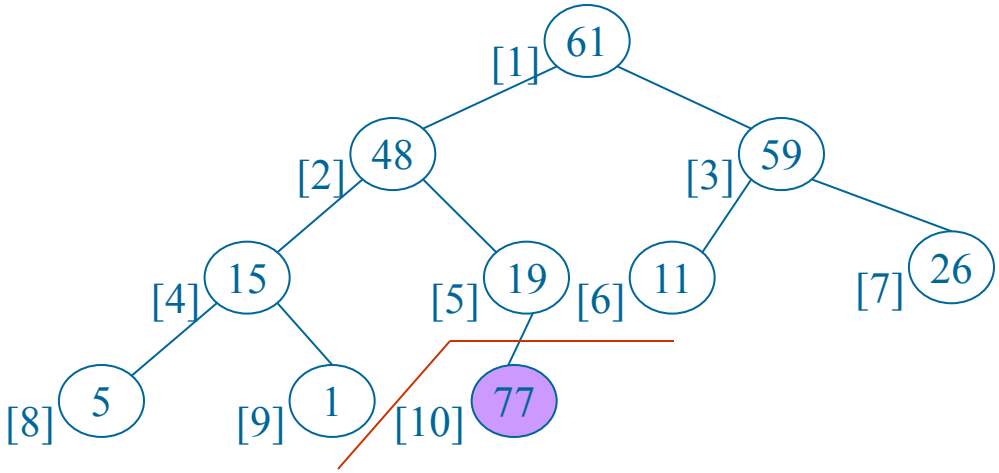
Mảng được tổ chức như cây nhị phân

1	2	3	4	5	6	7	8	9	10
26	5	77	1	61	11	59	15	48	19

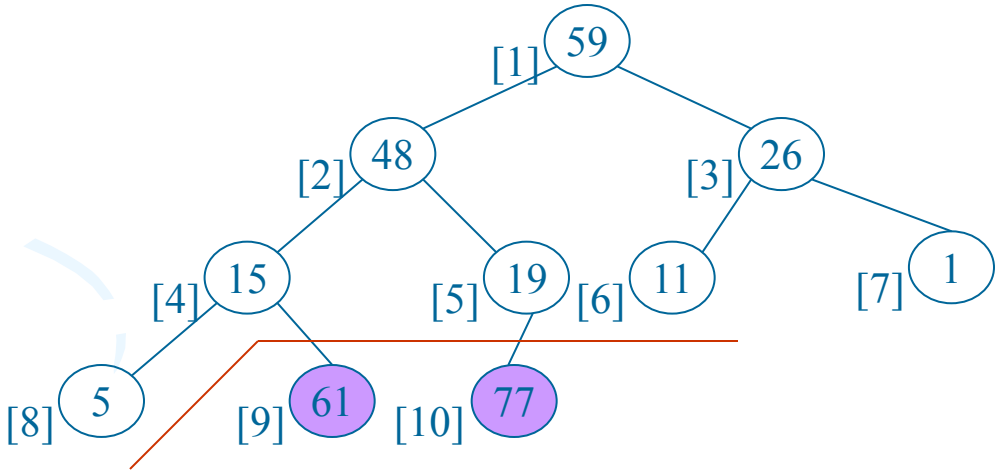
input file



Minh họa Heap sort

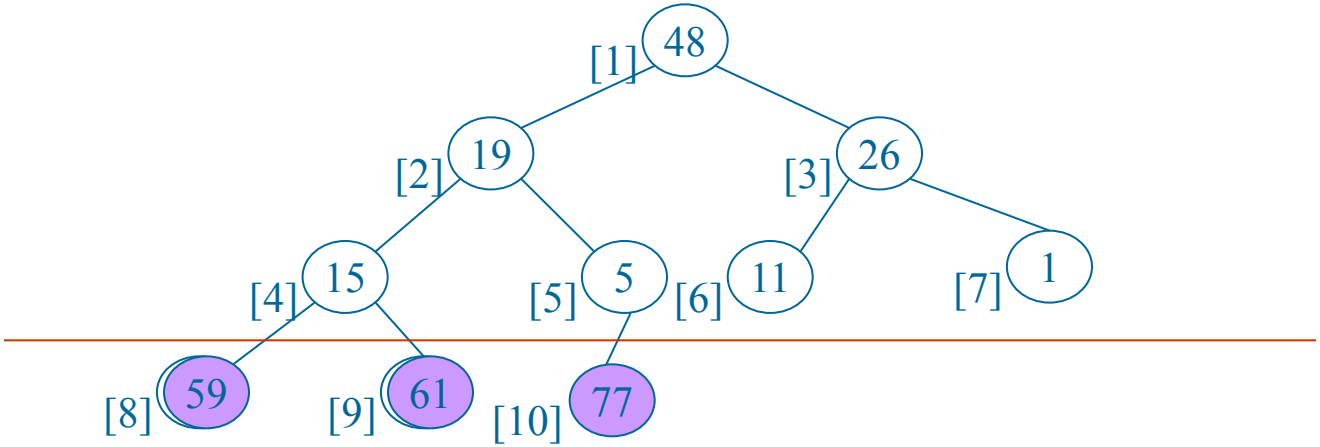


(a)

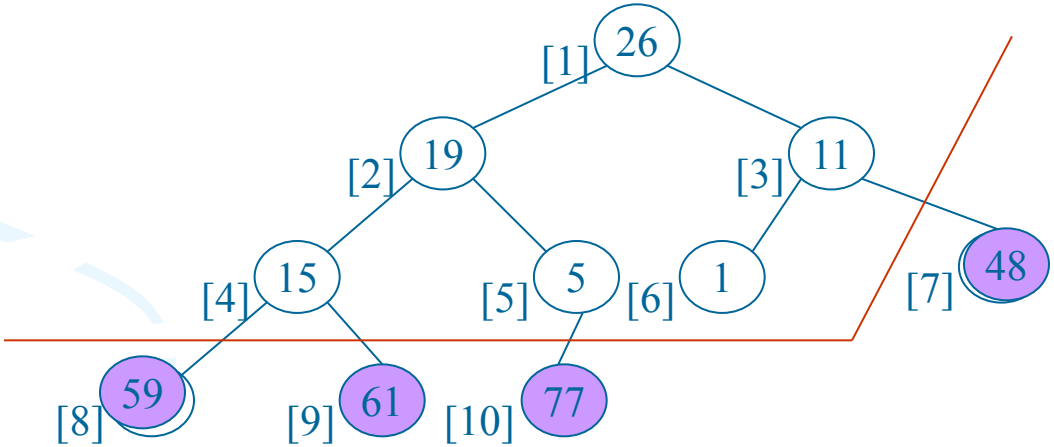


(b)

Minh họa illustration



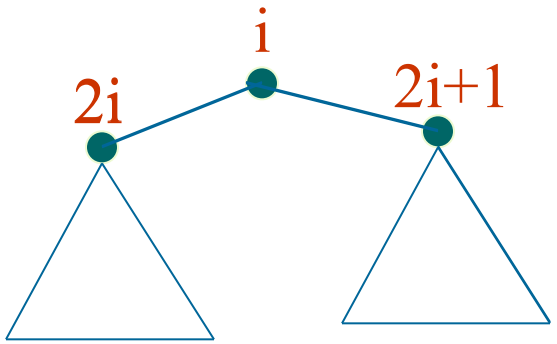
(c)



(d)

Heap sort

```
void adjust(element list[], int root, int n)
{
  int child, rootkey;   element temp;
  temp=list[root];      rootkey=list[root].key;
  child=2*root;
  while (child <= n) {
    if ((child < n) &&
        (list[child].key < list[child+1].key))
           child++;
    if (rootkey > list[child].key) break;
    else {
      list[child/2] = list[child];
      child *= 2;
    }
  }
  list[child/2] = temp;
}
```



Heap sort

```
void heapsort(element list[], int n)
{   //thứ tự tăng lên (đống cực đại)
    int i, j;
    element temp;
    for (i=n/2; i>0; i--) adjust(list, i, n);
    for (i=n-1; i>0; i--) {
        SWAP(list[1], list[i+1], temp);
        adjust(list, 1, i);
    }
}
```

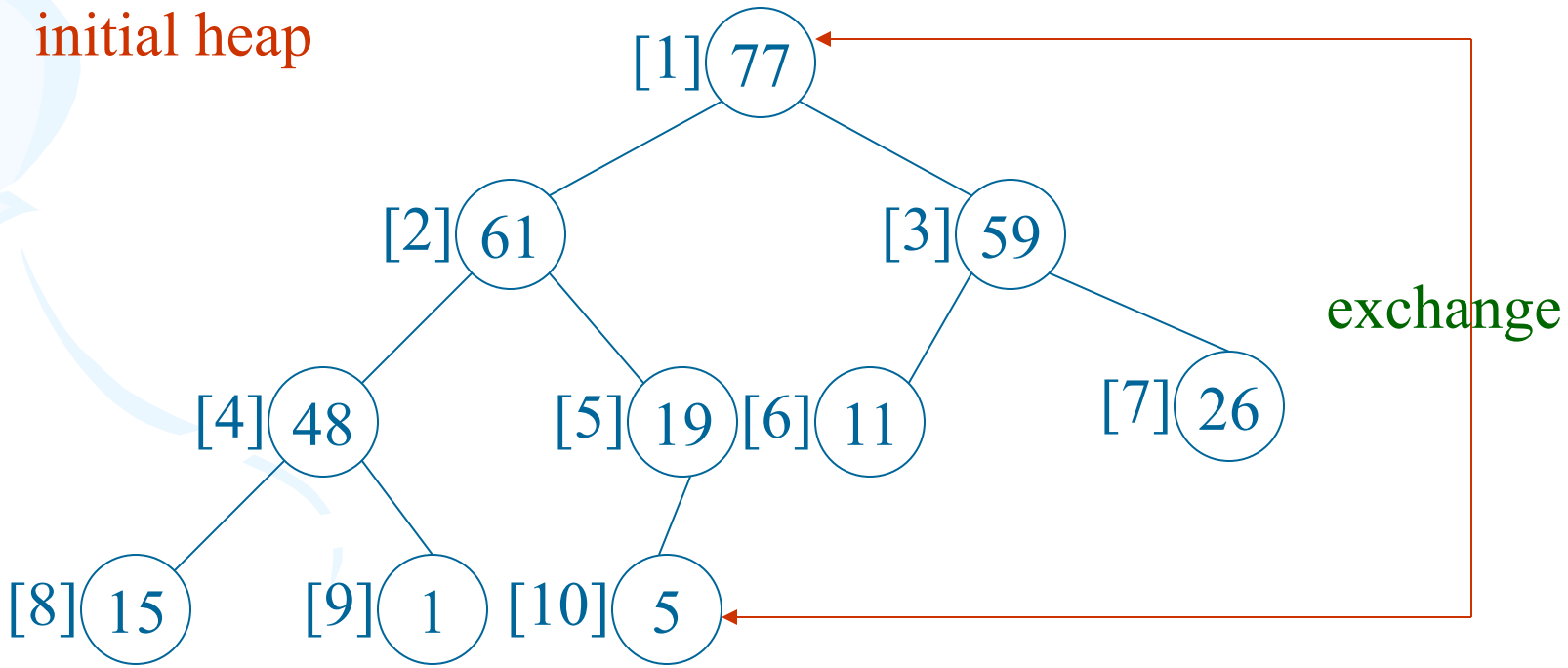
dưới lên

n-1 vòng

trên-xuống

Heap sort

Max heap following first **for** loop of *heapsort*



Exercise 2

- Tương tự bài tập 1 nhưng sử dụng heap sort.
- In ra số phép so sánh

Exercise 3: So sánh thời gian chạy

- Viết chương trình khởi tạo giá trị mảng 500 số nguyên bằng hàm random.
- Sắp xếp mảng bằng giải thuật sắp xếp chèn và vun đống.
- Tính thời gian chạy của chương trình theo từng trường hợp và in ra kết quả.

Help

- Hàm sinh số ngẫu nhiên: `srand(time(NULL))và rand()`
- Các hàm thời gian

```
#include <time.h>
time_t t1,t2;
time(&t1);
/* Làm việc bất kỳ*/
time(&t2);
durationinseconds = (int) t2 -t1;
```

Exercise 4

- Nhập 10 từ từ bàn phím và chuyển vào mảng các ký tự.
- Sắp xếp mảng theo sắp xếp chèn và xuất ra mảng đã sắp xếp ra màn hình.

Gợi ý

- Bạn có thể viết chương trình xử lý theo trình tự sau:
  - 1. Khai báo char data[10].
  - 2. Đọc từng từ 1 từ bàn phím bằng hàm fgetc( ) và nhập vào mảng "data".
  - 3. Sắp xếp trên mảng "data"
  - 4. Xuất từng từ 1 của mãng đã sắp xếp bằng hàm fputc( ).