

The background features a collection of abstract, colorful shapes including curved lines in shades of purple, blue, and green, and several yellow triangles of varying sizes scattered across the white space.

C Programming Basic – week 10

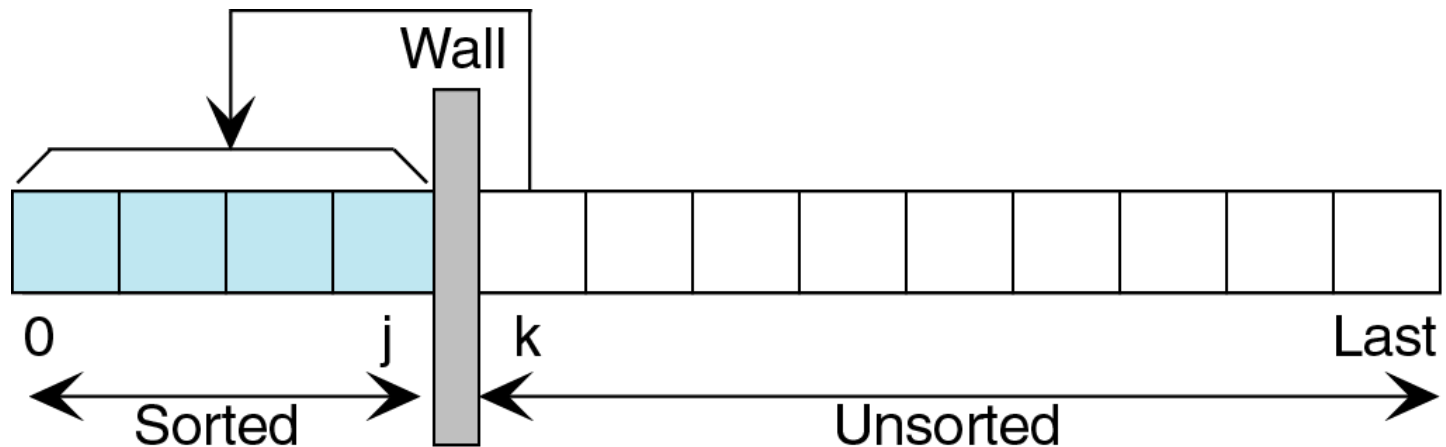
Nội dung

- Các giải thuật sắp xếp cơ bản
 - Sắp xếp chèn
 - Sắp xếp lựa chọn
- Sắp xếp vun đống

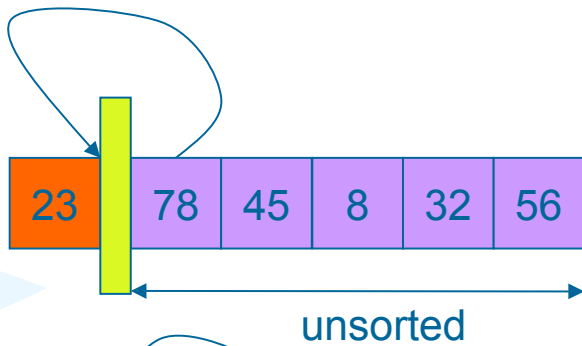
Sắp xếp chèn



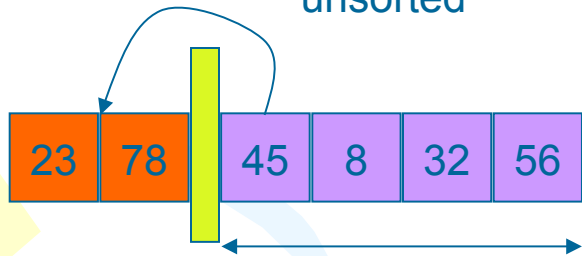
- Kỹ thuật xếp bài
- Quy tắc
 - Tìm phần tử chưa sắp xếp đầu tiên
 - Đưa nó đến vị trí đúng
 - Độ phức tạp: $O(n^2)$



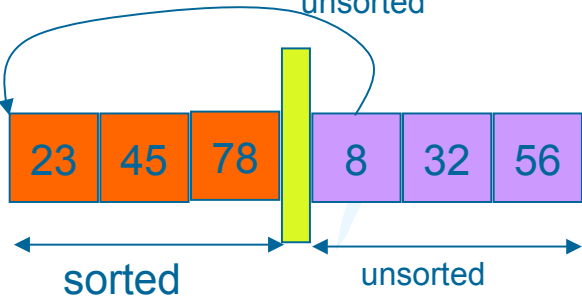
Original List



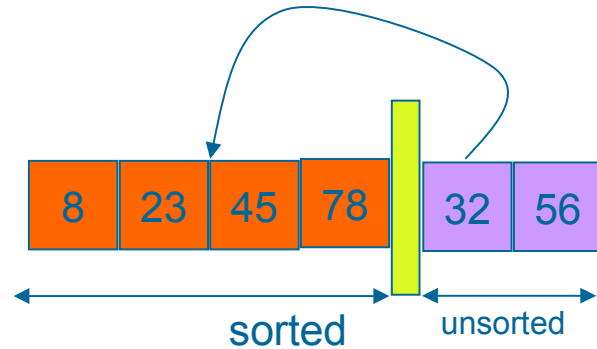
Apter step 1



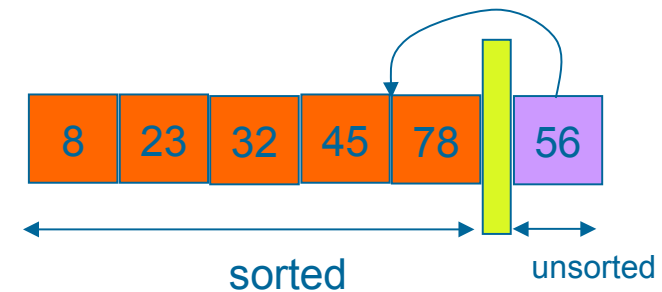
Apter step 2



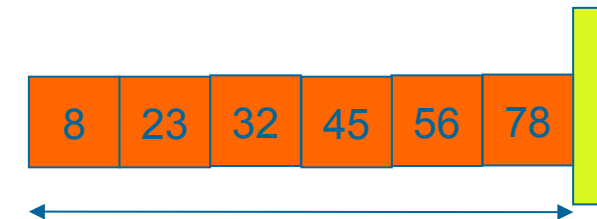
Apter step 3



Apter step 4



Apter step 5



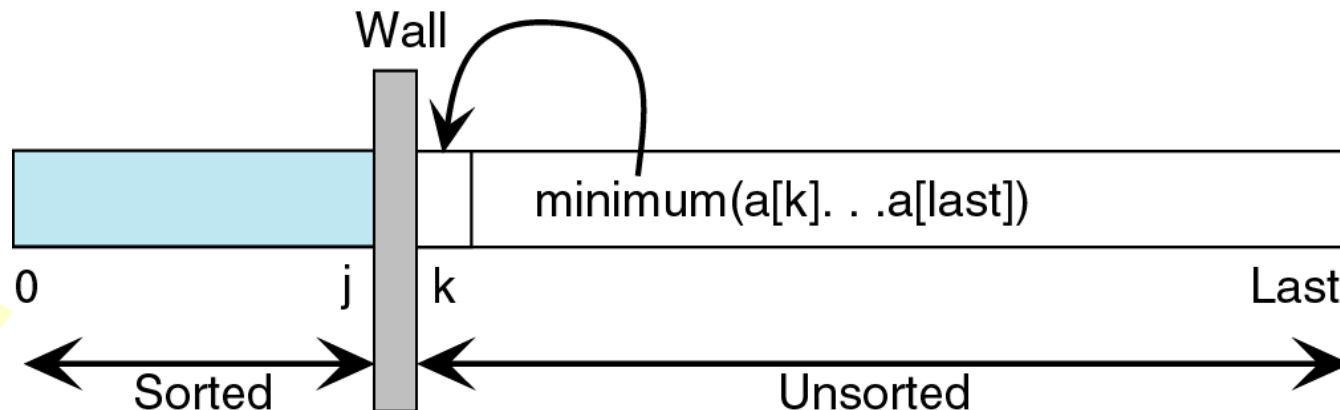
unsorted⁴

Insertion Sort

```
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 lựa chọn

- Quy tắc
 - Tìm phần tử bé nhất (lớn nhất) trong danh sách
 - Đưa nó lên đầu (cuối) danh sách bằng cách đổi chỗ với phần tử đầu (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;
  }
}
```

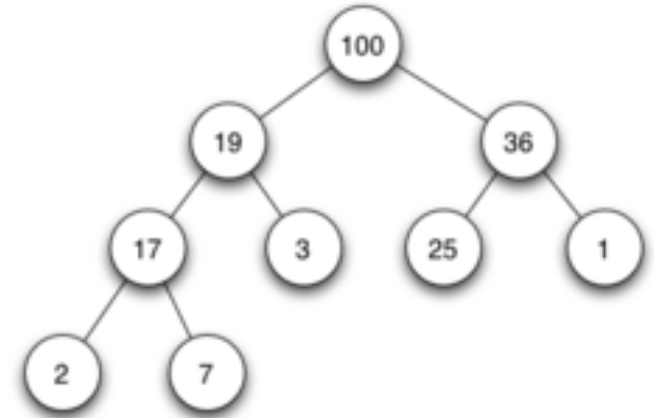
Exercise 10.1

- Xây dựng danh bạ điện thoại
- Khai báo cấu trúc với name, phone number, và email
- Đọc 10 bản ghi từ tệp vào danh sách, sắp xếp theo thứ tự tăng dần và ghi ra tệp
- Sử dụng sắp xếp chèn và sắp xếp lựa chọn

- (1) Sử dụng mảng các cấu trúc
- (2) Sử dụng danh sách liên kết đơn/đôi
- In ra số phép so sánh trong mỗi giải thuật

Sắp xếp vun đống

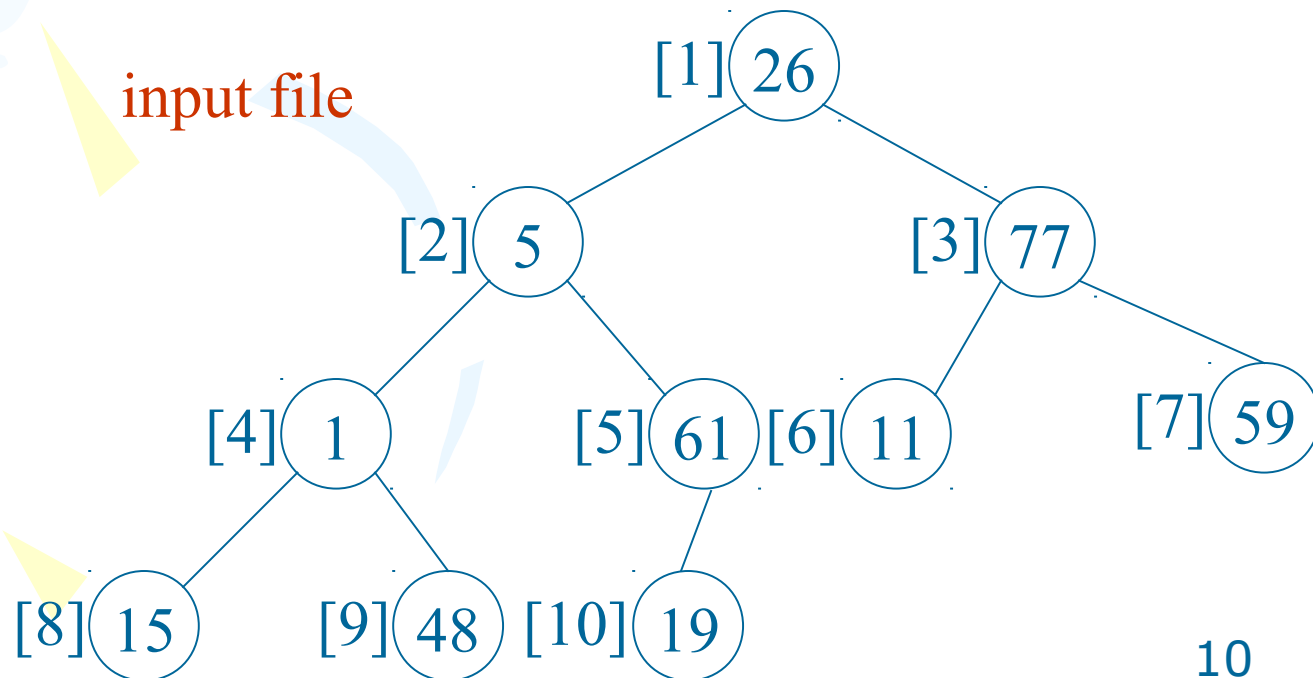
- Đống: cây nhị phân
 - Root chứa giá trị lớn nhất
 - Cây đầy đủ hoặc gần đầy đủ
 - Nút cha có giá trị lớn hơn các nút con



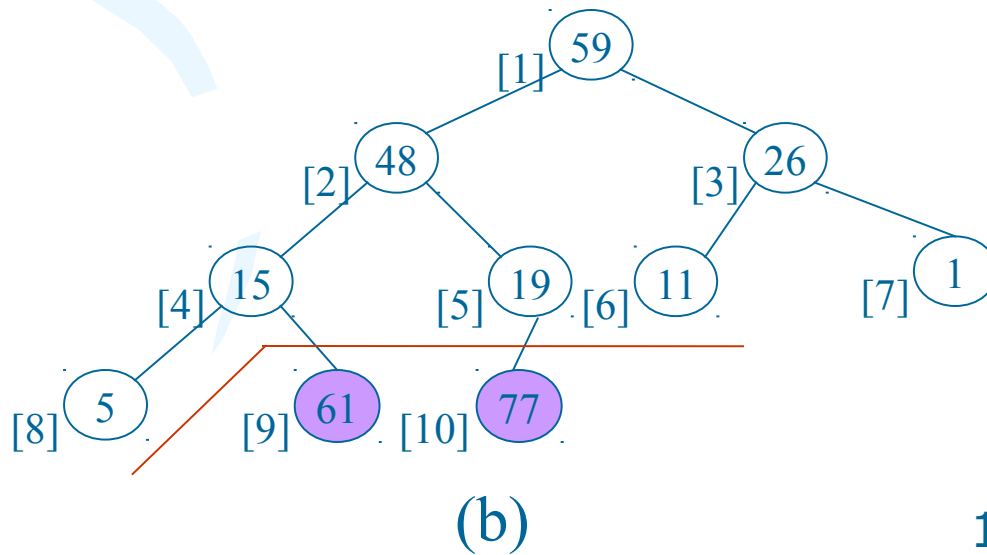
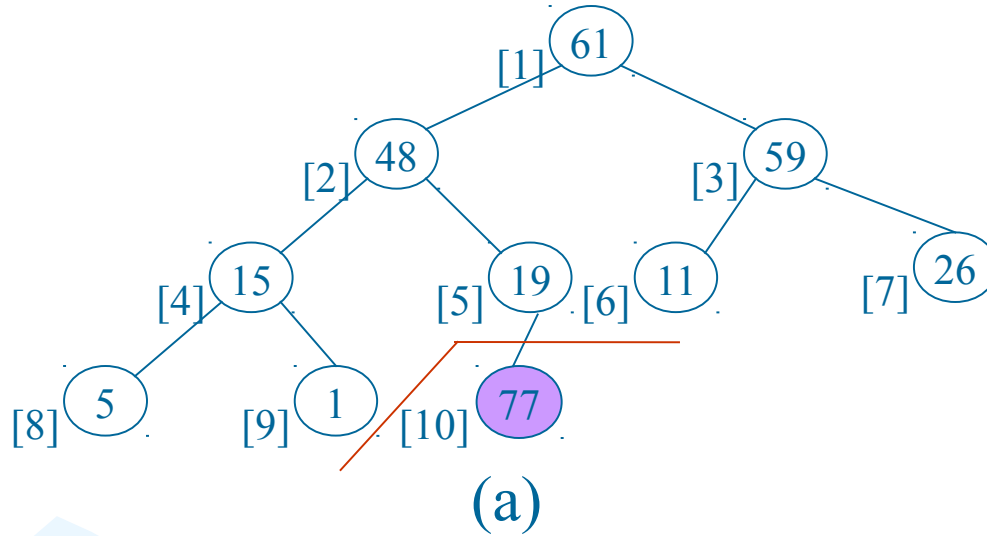
Sắp xếp vun đống (2)

Array interpreted as a binary tree

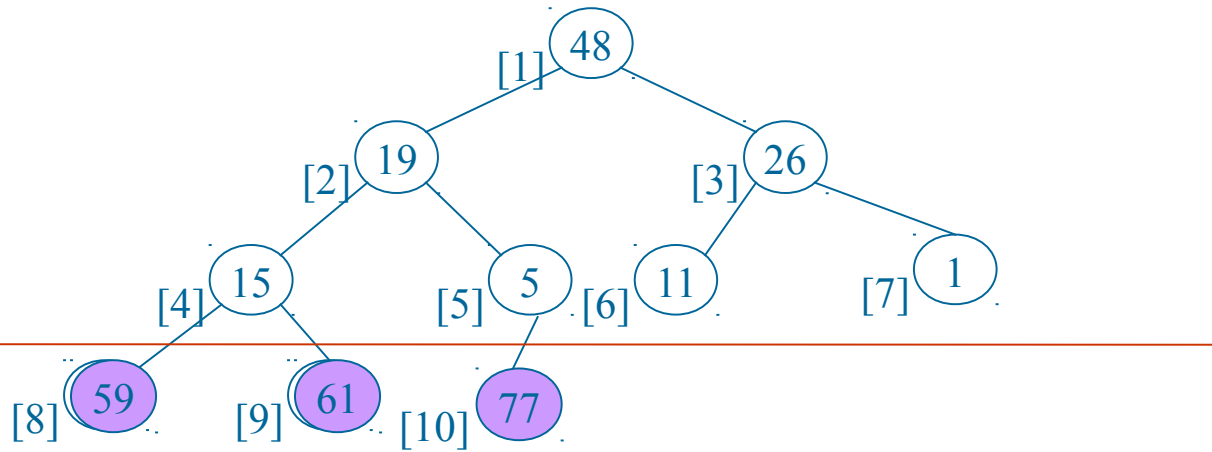
1	2	3	4	5	6	7	8	9	10
26	5	77	1	61	11	59	15	48	19



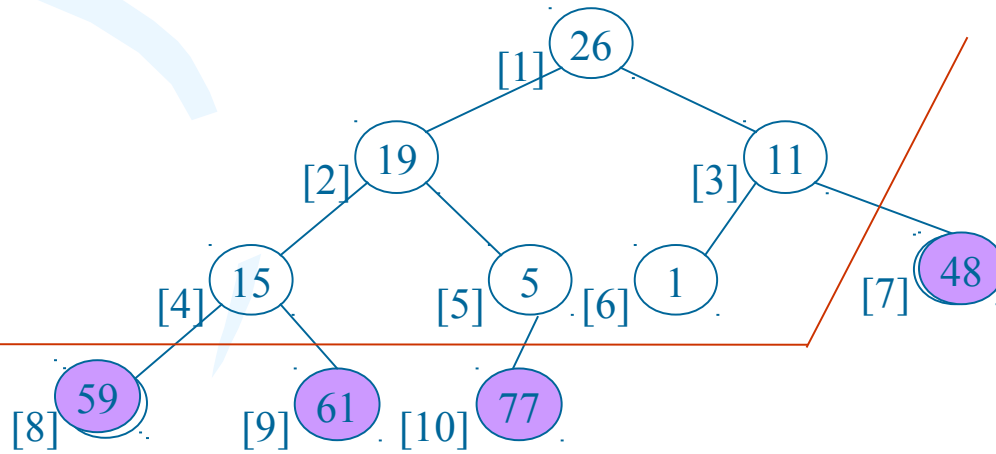
Minh họa



Minh họa (2)



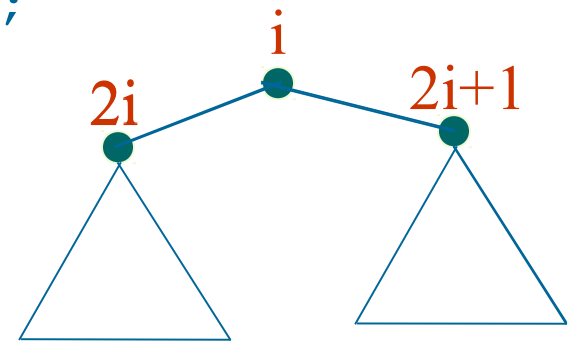
(c)



(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;
}
```

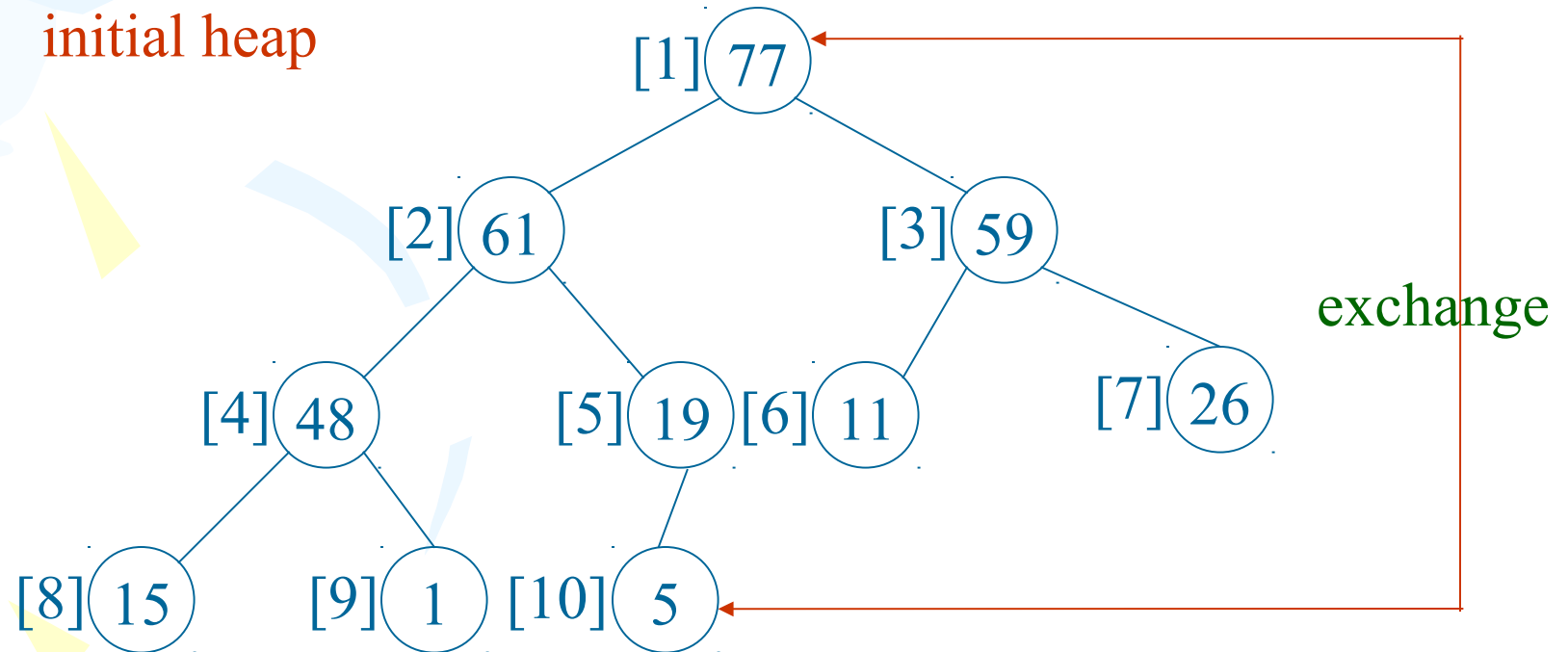


Heap sort (2)

```
void heapsort(element list[], int n)
{ ascending order (max heap)
  int i, j;
  element temp;
  for (i=n/2; i>0; i--) adjust(list, i,
                                bottom-up
                                n-1 cycles
                                n);
  for (i=n-1; i>0; i--) {
    SWAP(list[1], list[i+1], temp);
    top-down
    adjust(list, 1, i);
  }
}
```

Minh họa

Max heap following first for loop of *heapsort*



Exercise 10.2

- Xây dựng danh bạ điện thoại
- Viết chương trình có thể lưu trữ 100 cấu trúc chứa name, phone number, và email
- Đọc 10 bản ghi từ tệp, sắp xếp theo thứ tự tăng dần và in ra tệp
- Sử dụng sắp xếp vun đống. In ra số phép so sánh

Exercise 10.3

- Viết chương trình tạo một mảng ngẫu nhiên gồm 500 phần tử.
- Sắp xếp sử dụng sắp xếp chèn và sắp xếp vun đống. Tính thời gian thực hiện trong mỗi trường hợp và in ra kết quả.

Gợi ý

- Hàm sinh số ngẫu nhiên

`srand (time (NULL)) và rand ()`

- Hàm thời gian

```
#include <time.h>
```

```
time_t t1, t2;
```

```
time (&t1);
```

```
/* Do something */
```

```
time (&t2);
```

```
durationinseconds = (int) t2 - t1;
```

Exercise 10.4

- Nhập 10 từ từ bàn phím
- Với mỗi từ:
 - Lưu vào một mảng kí tự
 - Sắp xếp mảng bằng sắp xếp chèn và in kết quả ra màn hình

Gợi ý

- Giải thuật:
 - 1. Khai báo `char data[10]`.
 - 2. Đọc từ bằng hàm `fgetc()` vừa lưu vào mảng `data`
 - 3. Sắp xếp mảng `data`
 - 4. In ra màn hình bằng hàm `fputc()`

Homework 1

- Khai báo mảng 2 triệu số nguyên
- Sử dụng hàm ngẫu nhiên để khởi tạo mảng
- Viết chương trình dạng menu
 1. Tạo (lại) dữ liệu
 2. Sắp xếp chèn
 3. Sắp xếp lựa chọn
 4. Sắp xếp vun đống
- In ra thời gian thực hiện của mỗi giải thuật sắp xếp

Homework 2

- Từ tệp NokiaDB.dat, đọc dữ liệu, sắp xếp bằng sắp xếp vun đống và in ra màn hình