

The background features a white canvas with several abstract, colorful elements. There are thick, flowing lines in shades of light blue, light green, and light purple. Interspersed among these lines are numerous small, yellow triangles of varying sizes and orientations, some pointing towards the center and others towards the edges. The overall effect is a dynamic and celebratory pattern.

C Programming Basic – week 11

Nội dung

- Các giải thuật sắp xếp nâng cao
 1. Sắp xếp nhanh
 2. Mergesort
- Bài tập

1. Sắp xếp nhanh

Cho mảng n phần tử (vd số nguyên):

- If $n=1$, return
- Else
 - Chọn một phần tử làm *pivot*.
 - Chia mảng thành 2 mảng con
 - Các phần tử \geq pivot
 - Các phần tử $<$ pivot
 - Sắp xếp hai mảng con
 - Return kết quả

VD

- Cho mảng các số nguyên

40	20	10	80	60	50	7	30	100
----	----	----	----	----	----	---	----	-----

Quick Sort (Hoare)

- Given $(R_0, R_1, \dots, R_{n-1})$

K_i : pivot key

if K_i is placed in $S(i)$,

then $K_j \leq K_{S(i)}$ for $j < S(i)$,

$K_j \geq K_{S(i)}$ for $j > S(i)$.

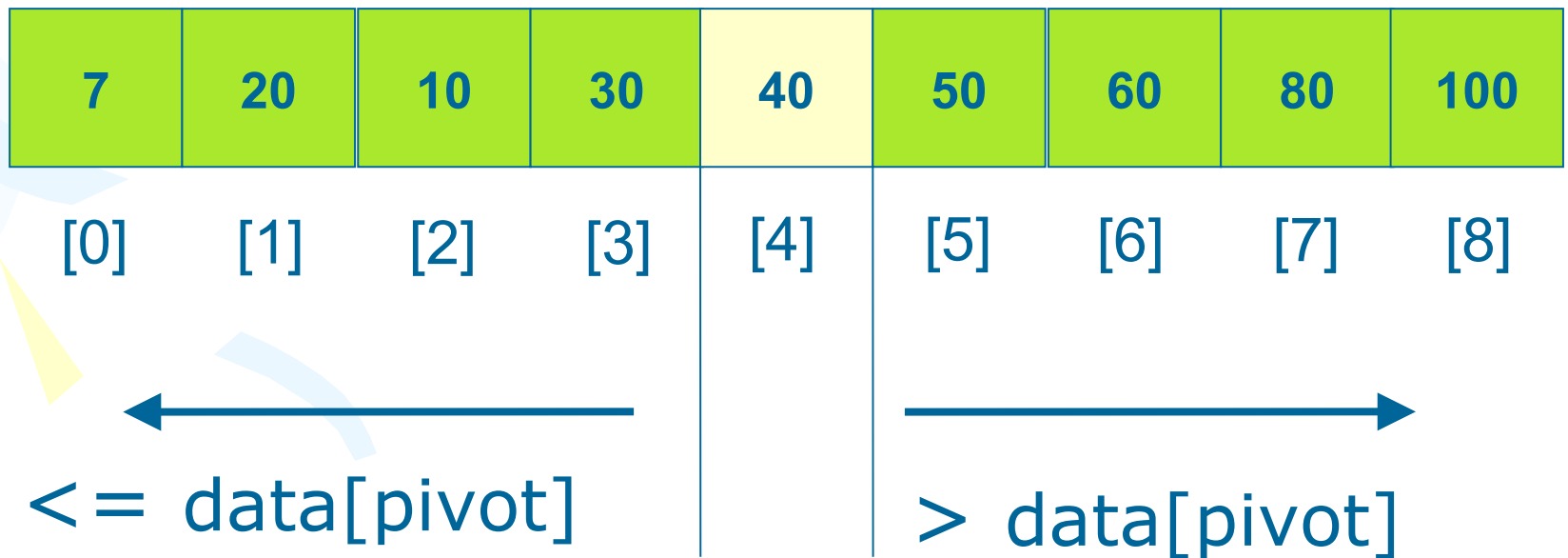
- $R_0, \dots, R_{S(i)-1}, R_{S(i)}, R_{S(i)+1}, \dots, R_{S(n-1)}$

two partitions

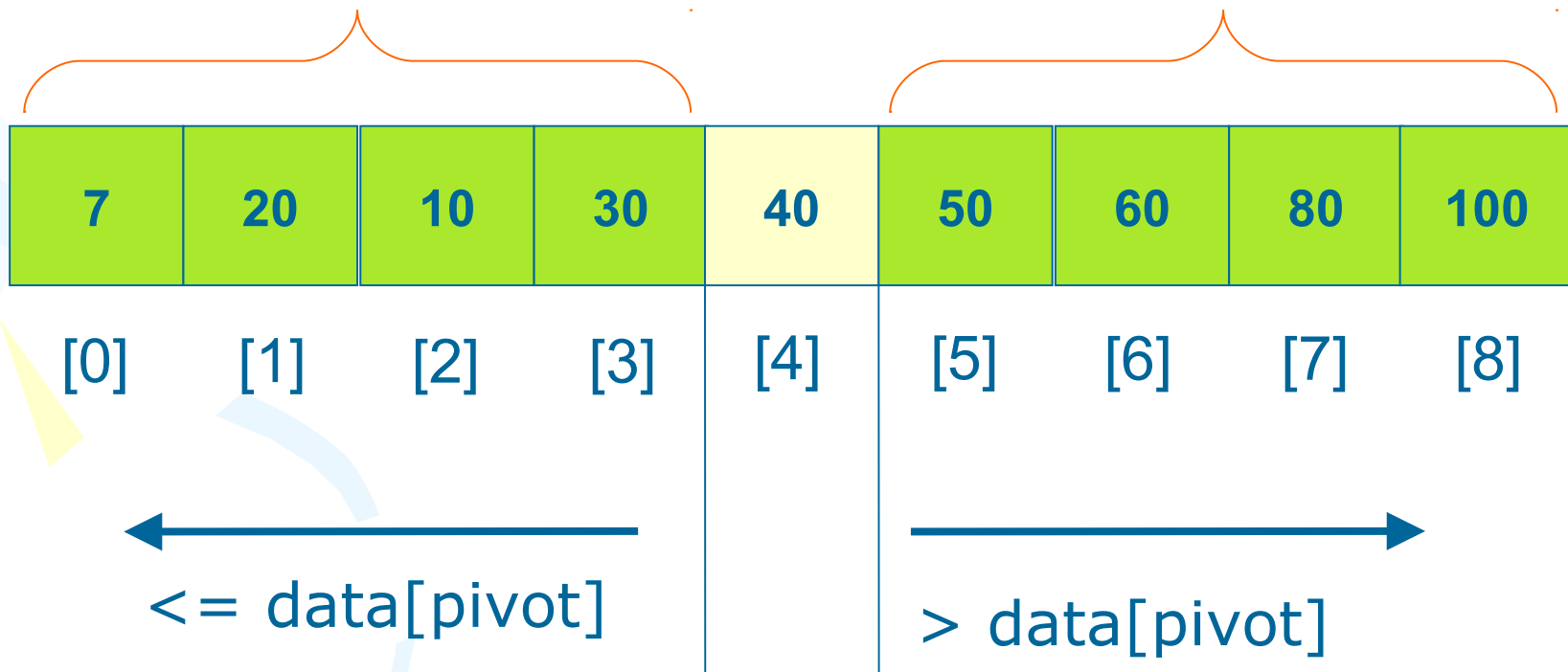
Phân vùng

- Cho một pivot, phân vùng sao cho mảng có:
 1. Một mảng con chứa các phần tử \geq pivot
 2. Mảng con còn lại chứa các phần tử $<$ pivot
- Hai mảng con được sắp xếp trực tiếp trên mảng gốc
- Phân vùng bằng cách đổi chỗ

VD phân vùng



Đệ quy



VD

R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	left	right
{ 26	5	37	1	61	11	59	15	48	19}	0	9
{ 11	5	19	1	15}	26	{ 59	61	48	37}	0	4
{ 1	5}	11	{ 19	15}	26	{ 59	61	48	37}	0	1
1	5	11	15	19	26	{ 59	61	48	37}	3	4
1	5	11	15	19	26	{ 48	37}	59	{ 61}	6	9
1	5	11	15	19	26	37	48	59	{ 61}	6	7
1	5	11	15	19	26	37	48	59	61	9	9
1	5	11	15	19	26	37	48	59	61		

Quick Sort

```
void quicksort(element list[], int left,
               int right)
{
    int pivot, i, j;
    element temp;
    if (left < right) {
        i = left;      j = right+1;
        pivot = list[left].key;
        do {
            do i++; while (list[i].key < pivot);
            do j--; while (list[j].key > pivot);
            if (i < j) SWAP(list[i], list[j], temp)
        } while (i < j);
        SWAP(list[left], list[j], temp);
        quicksort(list, left, j-1);
        quicksort(list, j+1, right);
    }
}
```

Exercise 11.1

- Xây dựng danh bạ điện thoại
- Định nghĩa cấu trúc chứa các thông tin "name", "phone number" và "e-mail address", khai báo mảng chứa 100 phần tử
- Viết chương trình đọc 10 bản ghi từ tệp, sắp xếp theo thứ tự tăng dần và ghi ra tệp. Sử dụng giải thuật sắp xếp nhanh

Exercise 11.2

- Khởi tạo một mảng gồm n số nguyên ngẫu nhiên. N được nhập bởi người dùng
- Sắp xếp mảng sử dụng sắp xếp chèn và sắp xếp nhanh
- So sánh thời gian thực hiện của hai giải thuật
- Chạy chương trình với các giá trị khác nhau của n và nhận xét

Exercise 11.3

- Viết chương trình lựa chọn giải thuật sắp xếp – Nếu số phần tử $> x$, chọn sắp xếp nhanh, nếu không chọn sắp xếp chèn
- Chú ý: x là một tham số của chương trình
- Đọc một tệp văn bản có nhiều hơn 100 kí tự, sắp xếp 100 kí tự đầu tiên, và in kết quả ra màn hình

2. Mergesort

- Phát biểu bài toán: Cho n phần tử, sắp xếp các phần tử theo thứ tự không giảm
- Áp dụng chiến thuật chia-đế-trị
 - If $n=1$ dừng
 - If $n>1$, phân vùng thành hai mảng con; sắp xếp các mảng con; kết hợp lại thành một mảng được sắp xếp

Giải thuật

MergeSort (E[0 .. N])

if $N < \text{threshold}$

InsertionSort (E[0..N])

else

copy E[0.. N/2] to U[0.. N/2]

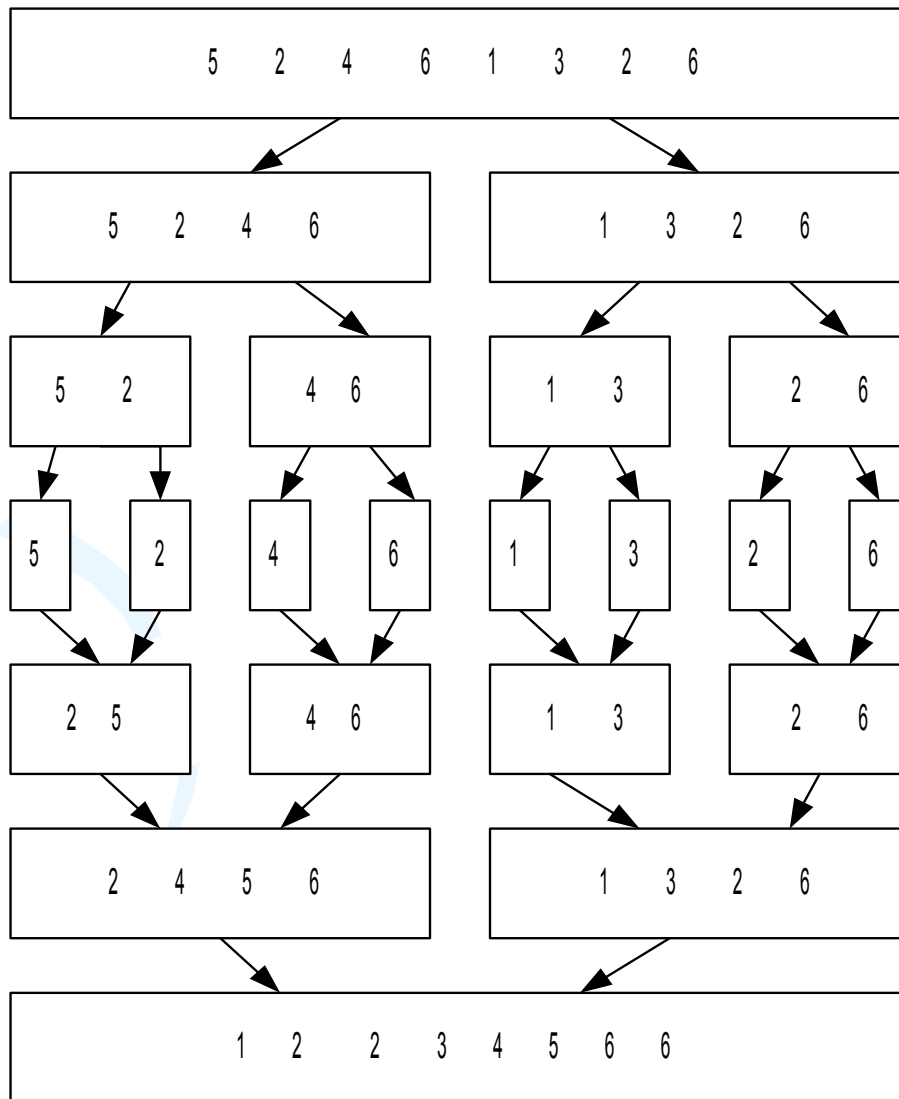
copy E[N/2 .. N] to V[0 .. N-N/2]

MergeSort(U[0 .. N/2])

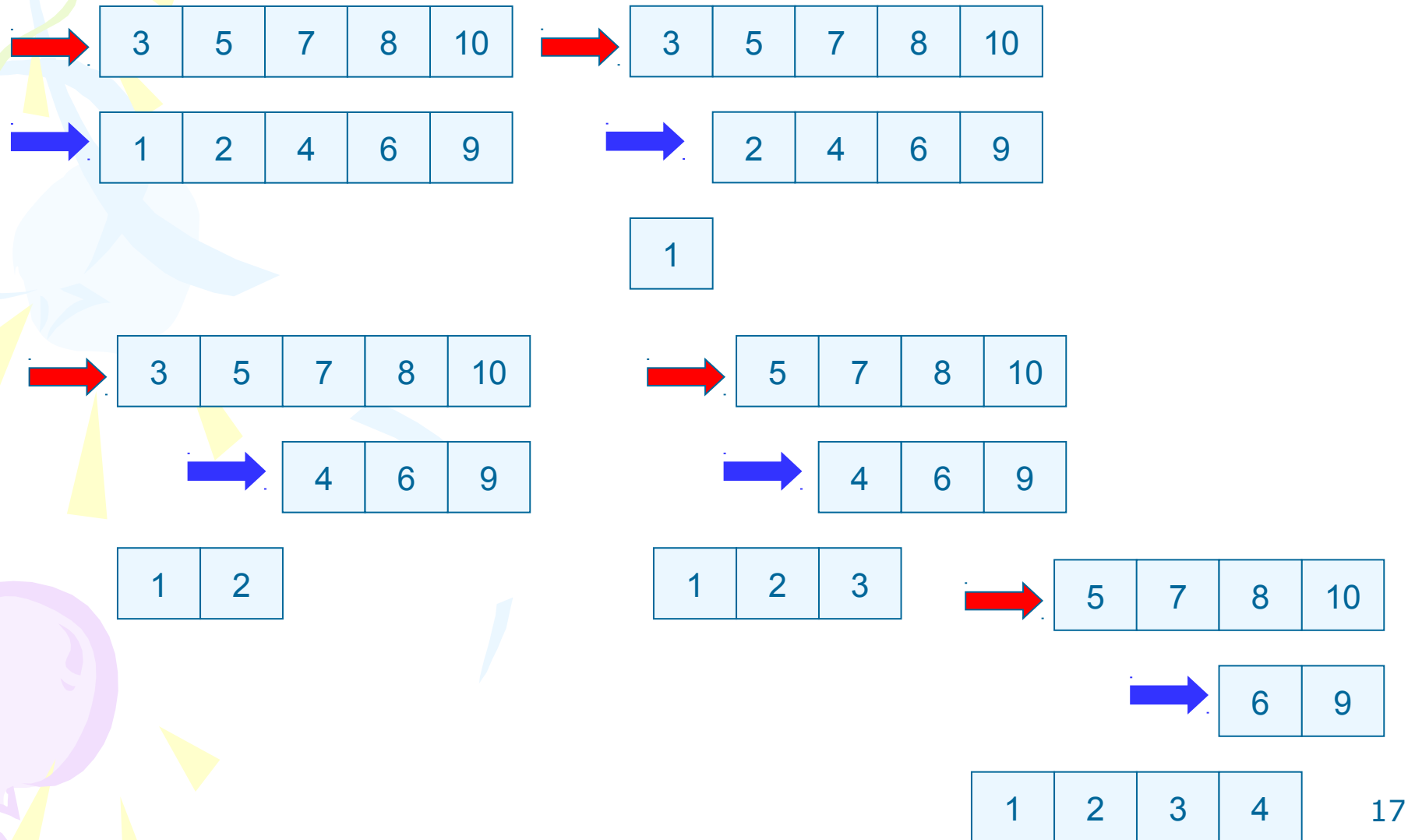
MergeSort(V[0 .. N-N/2])

*Merge(U[0 .. N/2], V[0 .. N-N/2},
E[0 .. N])*

VD



Mình họa



Giải thuật kết hợp

```
Merge (U[0..m] , V[0..n] , E[0..n+m] )
```

```
  i = 0 , j = 0
```

```
  k = 0
```

```
  while k < n+m
```

```
    if U[i] < V [j]
```

```
      E[k] = U[i] , i++
```

```
    else
```

```
      E[k] = V[j] , j++
```

```
    k++
```

Exercise 11.4

- Xây dựng danh bạ điện thoại
- Định nghĩa cấu trúc chứa các thông tin "name", "phone number" và "e-mail address", khai báo một danh sách liên kết đơn để lưu dữ liệu
- Viết chương trình đọc 10 bản ghi từ tệp, sắp xếp tăng dần và in ra màn hình. Sử dụng giải thuật mergesort.