



THUẬT TOÁN INSERTION SORT

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang



BÀI TOÁN DẪN NHẬP

Bài toán dẫn nhập



— Bài toán 1: Định nghĩa hàm thêm giá trị x vào mảng các số thực có k phần tử đã được sắp tăng sao cho mảng vẫn được sắp tăng.

— Ví dụ:

+ Mảng ban đầu:

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

+ Giá trị: $x = 42$.

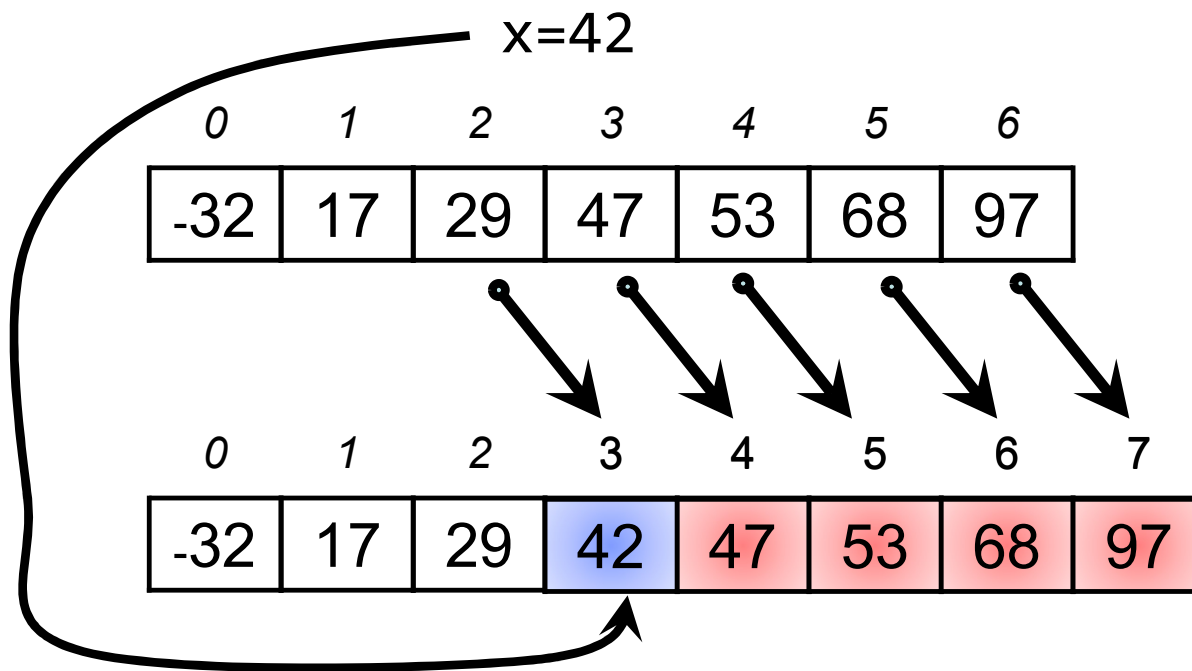
— Kết quả mảng sau khi chèn:

0	1	2	3	4	5	6	7
-32	17	29	42	47	53	68	97

Bài toán dẫn nhập



— Ví dụ:



Bài toán dẫn nhập



Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int k, float x)
```

```
12. {
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7

```
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7

```
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7

```
21. }
```


Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

```
13.     int vt = k-1;
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

```
21. }
```

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

```
13.     int vt = k-1;
```

```
14.     while( a[vt]>x)
```

```
15.     {
```

```
18.     }
```

```
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

```
13.     int vt = k-1;
```

```
14.     while( a[vt]>x)
```

```
15.     {
```

```
16.         a[vt+1] = a[vt];
```

```
18.     }
```

```
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

```
13.     int vt = k-1;
```

```
14.     while( a[vt]>x)
```

```
15.     {
```

```
16.         a[vt+1] = a[vt];
```

```
17.         vt--;
```

```
18.     }
```

```
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
```

```
12. {
```

```
13.     int vt = k-1;
```

```
14.     while(vt >= 0 && a[vt] > x)
```

```
15.     {
```

```
16.         a[vt+1] = a[vt];
```

```
17.         vt--;
```

```
18.     }
```

```
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7
-32	17	29	47	53	68	97	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	47	53	68	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6	
-32	17	29	47	53	68	97	
0	1	2	3	4	5	6	7
-32	17	29	47	53	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	47	47	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	47	47	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
19.     a[vt+1] = x;
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	47	47	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
19.     a[vt+1] = x;
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	42	47	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
19.     a[vt+1] = x;
20.     k++;
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	42	47	53	68	97



BÀI TOÁN DẪN NHẬP – CÀI ĐẶT GỌN

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     while(vt >= 0 && a[vt] > x)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
19.     a[vt+1] = x;
20.     k++;
21. }
```

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

$x=42$

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	42	47	53	68	97

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     int vt = k-1;
14.     for(; vt >= 0 && a[vt] > x;)
15.     {
16.         a[vt+1] = a[vt];
17.         vt--;
18.     }
19.     a[vt+1] = x;
20.     k++;
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13. |
14. |   for(int vt=k-1; vt>=0 && a[vt]>x;)
15. |   {
16. |       a[vt+1] = a[vt];
17. |       vt--;
18. |   }
19. |   a[vt+1] = x;
20. |   k++;
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13. |
14. |   for(int vt=k-1; vt>=0 && a[vt]>x; vt--)
15. |   {
16. |       a[vt+1] = a[vt];
17. |   }
18. |
19. |   a[vt+1] = x;
20. |   k++;
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13. |
14. |     for(int vt=k-1; vt>=0 && a[vt]>x; vt--)
15. |
16. |         a[vt+1] = a[vt];
17. |
18. |
19. |     a[vt+1] = x;
20. |     k++;
21. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     for(int vt=k-1; vt>=0 && a[vt]>x; vt--)
14.         a[vt+1] = a[vt];
15.     a[vt+1] = x;
16.     k++;
17. }
```

Bài toán dẫn nhập



```
11. void ThemBaoToan(float a[], int &k, float x)
12. {
13.     for(int vt=k-1; vt>=0 && a[vt]>x; vt--)
14.         a[vt+1] = a[vt];
15.     a[vt+1] = x;
16.     k++;
17. }
```

x=42

Thêm giá trị x vào mảng số thực có k phần tử đã được sắp tăng sao cho mảng vẫn tăng.

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

0	1	2	3	4	5	6	7
-32	17	29	42	47	53	68	97



VÀI CÂU HỎI GỢI Ý CHO THUẬT TOÁN

Vài câu hỏi gợi ý cho thuật toán



- Trước phần tử $a[0]$ có mấy phần tử?
- Trả lời: không có phần tử nào.
- Trước phần tử $a[1]$ có mấy phần tử?
- Trả lời: có 1 phần tử là $a[0]$.
- Trước phần tử $a[2]$ có mấy phần tử?
- Trả lời: có 2 phần tử là $a[0], a[1]$.
- Trước phần tử $a[i]$ có mấy phần tử?
- Trả lời: có i phần tử là $a[0], a[1], a[2], \dots, a[i-1]$.

0	1	2	3	4	5	6
29	97	68	17	-32	53	47



TƯ TƯỞNG THUẬT TOÁN

Tư tưởng thuật toán



- Thuật toán Insertion sort sắp xếp dựa trên tư tưởng là không gian cần sắp xếp đã được sắp xếp một phần và ta chỉ cần thêm một giá trị mới vào không gian này sao cho không gian mới được sắp xếp mà thôi.



THUẬT TOÁN INSERTION SORT

Thuật toán insertion sort



- Bước 1: Chèn $a[1]$ vào mảng a có 1 phần tử ($a[0]$) đã được sắp tăng để được mảng a có 2 phần tử sắp tăng ($a[0], a[1]$).
- Bước 2: Chèn $a[2]$ vào mảng a có 2 phần tử đã được sắp tăng để được mảng a có 3 phần tử sắp tăng ($a[0], a[1], a[2]$).
- ...
- Bước i : Chèn $a[i]$ vào mảng a có i phần tử đã được sắp tăng để được mảng a có $(i + 1)$ phần tử sắp tăng ($a[0], a[1], \dots, a[i]$).
- ...
- Bước $n - 1$: Chèn $a[n - 1]$ vào mảng a có $(n - 1)$ phần tử đã được sắp tăng để được mảng a có n phần tử sắp tăng.



CÀI ĐẶT THUẬT TOÁN

Cài đặt thuật toán



— Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.

— Ví dụ:

+ Mảng ban đầu:

0	1	2	3	4	5	6
29	97	68	17	-32	53	47

+ Kết quả mảng sau khi sắp:

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.     int k = 1;
14.     while(k <= n-1)
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.         k++;
21.     }
22. }
```



CHẠY TỪNG BƯỚC THUẬT TOÁN

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
89	11	56	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
89	11	56	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
89	11	56	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
89	11	56	91	5	16	21	56	23

11

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	89	89	56	91	5	16	21	56	23

11

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	89	89	56	91	5	16	21	56	23

11

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	11	89	56	91	5	16	21	56	23

11

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	89	56	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	89	56	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	89	56	91	5	16	21	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	89	89	91	5	16	21	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	89	89	91	5	16	21	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

91

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

91

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

5

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	5	16	21	56	23

5

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	91	91	16	21	56	23

5

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	89	89	91	16	21	56	23

5

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
11	56	56	89	91	16	21	56	23

5

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	11	11	56	89	91	16	21	56	23

5

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	11	11	56	89	91	16	21	56	23

5

Chạy từng bước thuật toán



	0	1	2	3	4	5	6	7	8
	5	11	56	89	91	16	21	56	23

5

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	91	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	91	16	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	91	16	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	91	16	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	91	91	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	89	89	91	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	56	89	91	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	56	56	89	91	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	21	56	23

16

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	21	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	21	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	21	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	91	91	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	89	89	91	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	56	89	91	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	56	56	89	91	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	56	23

21

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	56	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	56	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	91	91	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	89	91	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	89	89	91	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	23

56

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	23

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	23

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	91	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	89	89	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	56	89	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	56	89	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	56	56	56	89	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	23	56	56	89	91

23

Chạy từng bước thuật toán



0	1	2	3	4	5	6	7	8
5	11	16	21	23	56	56	89	91



CÀI ĐẶT THUẬT TOÁN

Cài đặt thuật toán



— Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.

— Ví dụ:

+ Mảng ban đầu:

0	1	2	3	4	5	6
29	97	68	17	-32	53	47

+ Kết quả mảng sau khi sắp:

0	1	2	3	4	5	6
-32	17	29	47	53	68	97

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.     int k = 1;
14.     while(k <= n-1)
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.         k++;
21.     }
22. }
```

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.     int k = 1;
14.     for(; k <= n-1; )
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.         k++;
21.     }
22. }
```

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.
14.     for(int k = 1; k <= n-1; )
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.         k++;
21.     }
22. }
```

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.
14.     for(int k = 1; k <= n-1; k++)
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.
21.     }
22. }
```

- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(int a[], int n)
12. {
13.     for(int k = 1; k <= n-1; k++)
14.     {
15.         int x = a[k];
16.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
17.             a[vt+1] = a[vt];
18.         a[vt+1] = x;
19.     }
20. }
```

Cài đặt thuật toán



- Bài toán: Định nghĩa hàm sắp mảng một chiều các số nguyên tăng dần bằng thuật toán Insertion sort.

```
11. void InsertionSort(int a[], int n)
12. {
13.     for(int k = 1; k <= n-1; k++)
14.     {
15.         int x = a[k];
16.         for(int vt = k-1; vt >= 0 && a[vt] > x; vt--)
17.             a[vt+1] = a[vt];
18.         a[vt+1] = x;
19.     }
20. }
```




Thuật toán Insertion sort và mảng cấu trúc

MẢNG CẤU TRÚC

Insertion sort và mảng cấu trúc



- Bài toán: Định nghĩa hàm sắp mảng một chiều các phân số tăng dần bằng thuật toán Insertion sort.
- Khai báo kiểu dữ liệu biểu diễn phân số.

```
11.struct phanso
12.{
13.|    int tu;
14.|    int mau;
15.};
16.typedef struct phanso PHANSO;
```

Insertion sort và mảng cấu trúc



```
11. int SoSanh(PHANSO x, PHANSO y)
12. {
13.     float a = (float)x.tu/x.mau;
14.     float b = (float)y.tu/y.mau;
15.     if(a>b)
16.         return 1;
17.     if(a<b)
18.         return -1;
19.     return 0;
20. }
```

- Bài toán: Định nghĩa hàm sắp mảng một chiều các phân số tăng dần bằng thuật toán Insertion sort.



```
11. void InsertionSort(PHANSO a[], int n)
12. {
13.     int k = 1;
14.     while(k <= n-1)
15.     {
16.         int x = a[k];
17.         for(int vt = k-1; vt >= 0 && SoSanh(a[vt], x) == 1; vt--)
18.             a[vt+1] = a[vt];
19.         a[vt+1] = x;
20.         k++;
21.     }
22. }
```

Insertion sort và mảng cấu trúc



- Bài toán: Định nghĩa hàm sắp mảng một chiều các phân số tăng dần bằng thuật toán Insertion sort.

```
11. void InsertionSort(PHANSO a[], int n)
12. {
13.     for(int k = 1; k <= n-1; k++)
14.     {
15.         int x = a[k];
16.         for(int vt = k-1; vt >= 0 && SoSanh(a[vt], x) == 1; vt--)
17.             a[vt+1] = a[vt];
18.         a[vt+1] = x;
19.     }
20. }
```



Thuật toán Insertion sort

ĐẶC ĐIỂM – ĐIỂM MẠNH – ĐIỂM YẾU

Đặc điểm – điểm mạnh – điểm yếu



— Đặc điểm thuật toán Insertion sort:

- + Độ phức tạp về thời gian (time complexity): $O(n^2)$.
- + Độ phức tạp về bộ nhớ (space complexity): $O(1)$.
- + Trường hợp xấu nhất (worst case): $O(n^2)$.
- + Trường hợp trung bình (average case): $O(n^2)$.
- + Trường hợp tốt nhất (best case): $O(n)$.
- + Ổn định.

Đặc điểm – điểm mạnh – điểm yếu



— Điểm mạnh:

- + Thuật toán rõ ràng, dễ hiểu.
- + Thuật toán dễ cài đặt.
- + Không yêu cầu dung lượng bộ nhớ lớn.
- + Hiệu quả trên bộ dữ liệu nhỏ.
- + Thuật toán chạy tốt khi dữ liệu sắp xếp một phần.
- + Thuật toán ổn định.
- + Hiệu quả hơn các thuật toán Selection, Bubble, Interchange.

Đặc điểm – điểm mạnh – điểm yếu



— Điểm yếu:

- + Thời gian thực hiện thuật toán lâu.
- + Không hiệu quả trên dữ liệu lớn.



Cảm ơn quý vị đã lắng nghe

ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM

TOÀN DIỆN – SÁNG TẠO – PHỤNG SỰ