

Chương 2

Phân tích độ phức tạp của Insertion Sort

Nội dung

1. Giới thiệu
2. Ý tưởng của thuật toán
3. Ví dụ minh họa
4. Minh họa bằng code java/C++
5. Đánh giá độ phức tạp của thuật toán

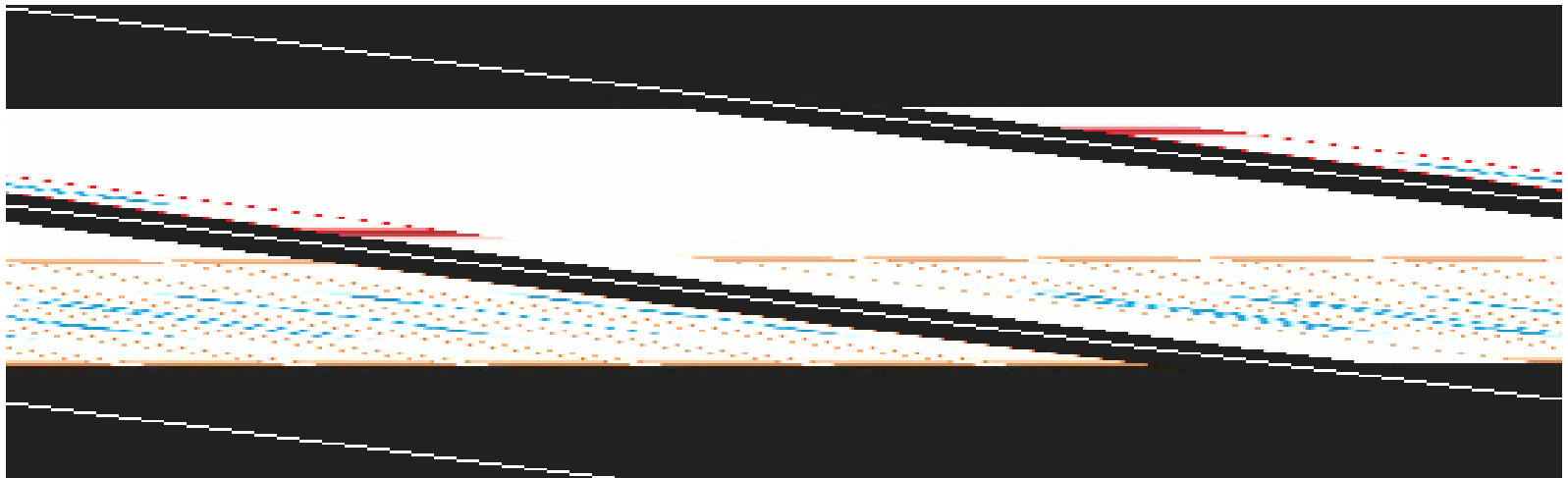
1. Giới thiệu

- Giới thiệu với các em một thuật toán sắp xếp đơn giản, dễ hiện thực và sử dụng.
- Insertion Sort – hay gọi là sắp xếp chèn.
- Ví dụ ta hiện thực việc sắp xếp tăng dần, có thể làm tương tự cho việc sắp xếp giảm dần và sử dụng mảng các số nguyên để minh họa cho giải thuật.

2. Ý tưởng của thuật toán

- Xét danh sách con gồm k phần tử đầu $a_1 \dots a_k$.
- Với $k = 1$, danh sách gồm một phần tử đã được sắp xếp thành dãy tăng dần.
- Giả sử trong danh sách $k-1$ phần tử đầu $a_1 \dots a_{k-1}$ đã được sắp xếp.
- Để sắp xếp phần tử $a_k = x$ ta tìm vị trí thích hợp của nó trong dãy $a_1 \dots a_{k-1}$.
- Vị trí thích hợp cần tìm là vị trí đứng trước phần tử lớn hơn nó và sau phần tử nhỏ hơn hoặc bằng nó.

3. Ví dụ minh họa



3. Ví dụ minh họa



$16 > 5$, shift

$9 > 5$, shift

$7 > 5$, shift

$3 < 5 < 7$, shifting is done

insert 5 to final position

4. Minh họa thuật toán

```
3 public class InsertionSort {
4     public static void insertionSort(int array[]) {
5         int n = array.length;
6         for (int j = 1; j < n; j++) {
7             int key = array[j];
8             int i = j-1;
9             while ( (i > -1) && ( array [i] > key ) ) {
10                 array [i+1] = array [i];
11                 i--;
12             }
13             array[i+1] = key;
14         }
15     }
16 }
```

4. Minh họa thuật toán

```
17 public static void main(String a[]){
18     int[] arr1 = {9,14,3,2,43,11,58,22};
19     System.out.println("Before Insertion Sort");
20     for(int i:arr1){
21         System.out.print(i+" ");
22     }
23     System.out.println();
24
25     insertionSort(arr1);//sorting array using insertion sort
26
27     System.out.println("After Insertion Sort");
28     for(int i:arr1){
29         System.out.print(i+" ");
30     }
31 }
```

Problems @ Javadoc Declaration Console

<terminated> InsertionSort [Java Application] C:\Program File

Before Insertion Sort
9 14 3 2 43 11 58 22
After Insertion Sort
2 3 9 11 14 22 43 58

5. Đánh giá thuật toán

```
3 public class InsertionSort {  
4     public static void insertionSort(int array[]) {  
5         int n = array.length;  
6         for (int j = 1; j < n; j++) {  
7             int key = array[j];  
8             int i = j-1;  
9             while ( ( i > -1 ) && ( array [i] > key ) ) {  
10                 array [i+1] = array [i];  
11                 i--;  
12             }  
13             array[i+1] = key;  
14         }  
15     }  
16 }
```

$$T(j) = 4 + 6*i$$

$$j = 1 \rightarrow n$$

$$T(n) = (4 + 6*1) + (4 + 6*2) + \dots + (4 + 6*k) + \dots + (4 + 6*n)$$

$$= 4*n + 6(1+2+\dots+k+\dots+n)$$

$$= 4*n + 6n(n+1)/2$$

$$= 4*n + (6n^2)/2 + 6n/2$$

$$= (6n^2)/2 + 14n/2$$

$$\Rightarrow O(n^2)$$

THANK YOU