



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM
MÔN: **KỸ THUẬT LẬP TRÌNH**

HƯỚNG DẪN THỰC HÀNH

TUẦN 08

BÀI TOÁN SẮP XẾP

TP.HCM, ngày 20 tháng 02 năm 2023

MỤC LỤC

1	Giới thiệu	3
2	Thuật toán sắp xếp chọn (Selection Sort).....	3
3	Thuật toán sắp xếp đổi trực tiếp (Interchange Sort)	4
4	Thuật toán sắp xếp chèn (Insertion Sort).....	4
5	Thuật toán sắp xếp nổi bọt (Bubble Sort).....	5
6	Thuật toán sắp xếp rung lắc (Shake Sort)	5
7	Bài tập.....	6

1 Giới thiệu

Bài toán sắp xếp là một trong những bài toán phổ biến nhất trong khoa học máy tính. Chúng ta sẽ xem qua một vài thuật toán sắp xếp cơ bản trên cấu trúc mảng. Để thuận tiện trong việc trình bày, chúng ta sẽ mặc định mảng được yêu cầu sắp xếp tăng.

2 Thuật toán sắp xếp chọn (Selection Sort)

Ý tưởng chính của thuật toán này là tìm ra chỉ số của phần tử có giá trị nhỏ nhất trong mảng. Sau đó sẽ chuyển phần tử này về phía bên trái nhất của mảng.

Dòng	
1	<code>void swap(int* a, int* b){</code>
2	<code> int temp = *a;</code>
3	<code> *a = *b;</code>
4	<code> *b = temp;</code>
5	<code>}</code>
6	
7	<code>void SelectionSort(int a[], int n){</code>
8	<code> int pos;</code>
9	<code> for(int i = 0; i < n - 1; i++){</code>
10	<code> pos = i;</code>
11	<code> for(int j = i + 1; j < n; j++){</code>
12	<code> if(a[j] < a[pos]) pos = j;</code>
13	<code> swap(&a[i], &a[pos]);</code>
14	<code> }</code>
15	<code>}</code>

Bảng bên dưới diễn tả quá trình chạy từng bước của thuật toán này

Lần chạy	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
0	9	6	3	5	8	2	1	4
1	1	6	3	5	8	2	9	4
2	1	2	3	5	8	6	9	4
3	1	2	3	5	8	6	9	4
4	1	2	3	4	8	6	9	5
5	1	2	3	4	5	6	9	8
6	1	2	3	4	5	6	9	8
7	1	2	3	4	5	6	8	9

3 Thuật toán sắp xếp đổi trực tiếp (Interchange Sort)

Ý tưởng chính của thuật toán này hoàn toàn tương tự như Selection Sort, có điều khi phát hiện phần tử bên phải nhỏ hơn phần tử bên trái thì thuật toán đổi chỗ ngay chứ không phải chỉ lưu lại chỉ số như trong Selection Sort.

Dòng	
1	<code>void InterchangeSort(int a[], int n){</code>
2	<code>for(int i = 0; i < n - 1; i++)</code>
3	<code>for(int j = i + 1; j < n; j++)</code>
4	<code>if(a[j] < a[i])</code>
5	<code>swap(&a[i], &a[j]);</code>
6	<code>}</code>

Bảng bên dưới diễn tả quá trình chạy từng bước của thuật toán này

Lần chạy	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
0	9	6	3	5	8	2	1	4
1	1	9	6	5	8	3	2	4
2	1	2	9	6	8	5	3	4
3	1	2	3	9	8	6	5	4
4	1	2	3	4	9	8	6	5
5	1	2	3	4	5	9	8	6
6	1	2	3	4	5	6	9	8
7	1	2	3	4	5	6	8	9

4 Thuật toán sắp xếp chèn (Insertion Sort)

Ý tưởng của thuật toán này là giả sử phía bên trái của mảng đã được sắp, tiếp đó ta sẽ duyệt lần lượt từng phần tử phía bên phải của mảng rồi tìm vị trí thích hợp phía bên trái mảng để chèn vào.

Dòng	
1	<code>void InsertionSort(int a[], int n){</code>
2	<code>int x;</code>
3	<code>for(int i = 1; i < n; i++){</code>
4	<code>x = a[i]; j = i - 1;</code>
5	<code>while((j >= 0) && (a[j] > x)){</code>
6	<code>a[j + 1] = a[j]; j--;</code>
7	<code>}</code>
8	<code>a[j + 1] = x;</code>
9	<code>}</code>
10	<code>}</code>

Bảng bên dưới diễn tả quá trình chạy từng bước của thuật toán này

Lần chạy	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
0	9	6	3	5	8	2	1	4
1	6	9	3	5	8	2	1	4
2	3	6	9	5	8	2	1	4
3	3	5	6	9	8	2	1	4
4	3	5	6	8	9	2	1	4
5	2	3	5	6	8	9	1	4
6	1	2	3	5	6	8	9	4
7	1	2	3	4	5	6	8	9

5 Thuật toán sắp xếp nổi bọt (Bubble Sort)

Ý tưởng thuật toán này khá tương tự với Interchange Sort nhưng thay vì xử lí ở đầu mảng, ta xử lí ở cuối. Và phần tử nhỏ nhất sẽ được **NỔI** lên đầu mảng.

Dòng	
1	<code>void BubbleSort(int a[], int n){</code>
2	<code> for(int i = 1; i < n; i++)</code>
3	<code> for(int j = n - 1; j >= i; j--)</code>
4	<code> if(a[j - 1] > a[j])</code>
5	<code> swap(&a[j - 1], &a[j]);</code>
6	<code> }</code>

Bảng bên dưới diễn tả quá trình chạy từng bước của thuật toán này

Lần chạy	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
0	9	6	3	5	8	2	1	4
1	1	9	6	3	5	8	2	4
2	1	2	9	6	3	5	8	4
3	1	2	3	9	6	4	5	8
4	1	2	3	4	9	6	5	8
5	1	2	3	4	5	9	6	8
6	1	2	3	4	5	6	9	8
7	1	2	3	4	5	6	8	9

6 Thuật toán sắp xếp rung lắc (Shake Sort)

Ý tưởng của thuật toán này là tận dụng ưu điểm của Bubble Sort đó là phần tử nhỏ nhất trở về đầu mảng và của Interchange Sort đó là phần tử lớn nhất trở về cuối mảng ngay từ vòng lặp đầu tiên.

Dòng	
1	<code>void ShakeSort(int a[], int n){</code>
2	<code>int left = 1, right = n - 1, j, k;</code>
3	<code>do{</code>
4	<code>for(j = right; j >= left; j--)</code>
5	<code>if(a[j - 1] > a[j]){</code>
6	<code>swap(&a[j], &a[j - 1]);</code>
7	<code>k = j;</code>
8	<code>}</code>
9	<code>left = k + 1;</code>
10	<code>for(j = left; j <= right; j++)</code>
11	<code>if(a[j - 1] > a[j]){</code>
12	<code>swap(&a[j], &a[j - 1]);</code>
13	<code>k = j;</code>
14	<code>}</code>
15	<code>right = k - 1;</code>
16	<code>}while(left <= right);</code>
17	<code>}</code>

Bảng bên dưới diễn tả quá trình chạy từng bước của thuật toán này

Lần chạy	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
0	9	6	3	5	8	2	1	4
1	1	6	3	5	8	2	4	9
2	1	2	3	5	6	4	8	9
3	1	2	3	4	5	6	8	9

7 Bài tập

Sinh viên xây dựng các hàm sắp xếp theo yêu cầu

- Sắp xếp mảng số nguyên theo thứ tự giảm dần
- Sắp xếp các phần tử chẵn trong mảng số nguyên theo thứ tự giảm dần
- Sắp xếp các phần tử âm thì tăng và dương thì giảm trong mảng số thực
- Sắp xếp các phần tử nguyên tố trong mảng số nguyên theo thứ tự tăng dần
- Hàm sắp xếp mảng một chiều các số nguyên theo yêu cầu sau:
 - + Tất cả các số chẵn dồn về phía trái của mảng theo thứ tự tăng dần
 - + Tất cả các số lẻ dồn về phía phải của mảng theo thứ tự giảm dần

Sinh viên tự thiết kế hàm main minh họa **TẤT CẢ** các **hàm sắp xếp chính** bên trên.