**2.2** Dùng phương pháp xếp thứ tự SelectionSort, mô tả từng bước quá trình xếp thứ tự dãy số trên (không lập trình). Tín độ phức tạp của quá trình xếp thứ tự danh sách trên.

Ta có danh sách gồm 8 phần tử không có thứ tự:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 70 | 20 | 60 | 90 | 10 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Từ vị trí đầu tiên trong danh sách đã được sắp xếp, toàn bộ danh sách được duyệt một cách liên tục. Vị trí đầu tiên có giá trị 40, chúng ta tìm toàn bộ danh sách và thấy rằng 10 là giá trị nhỏ nhất.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 70 | 20 | 60 | 90 | 10 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

.

Do đó, chúng ta thay thế 40 với 10. Sau một vòng lặp, giá trị 10 thay thế cho giá trị 40 tại vị trí đầu tiên trong danh sách đã được sắp xếp. Ta tráo đổi hai giá trị này. (1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 70 | 20 | 60 | 90 | 40 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Tại vị trí thứ hai, giá trị 70, chúng ta tiếp tục quét phần còn lại của danh sách theo thứ tự từng phần tử.

Chúng ta thấy rằng 20 là giá trị nhỏ nhất thứ hai trong danh sách và nó nên xuất hiện ở vị trí thứ hai. Chúng ta tráo đổi hai giá trị này.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 70 | 20 | 60 | 90 | 40 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 70 | 60 | 90 | 40 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(2)

Sau hai vòng lặp, hai giá trị nhỏ nhất đã được đặt tại phần đầu của danh sách đã được sắp xếp. Ta tiếp tục với vị trí thứ ba và duyệt tiếp tục danh sách chưa sắp xếp.

Ta tìm được vị trí có giá trị 30 là giá trị nhỏ nhất chưa được sắp xếp, nên tiếp tục hoán đổi vị trí của giá trị 70 và 30.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 70 | 60 | 90 | 40 | 50 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 60 | 90 | 40 | 50 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(3)

Tiếp tục với các vị trí còn lại, ta được danh sách với phần thử nhỏ nhất thứ tư với giá trị 40 và hoán vị 2 vị trí thì được danh sách:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 60 | 90 | 40 | 50 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 90 | 60 | 50 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(4)

Tiến trình tương tự sẽ được áp dụng cho phần còn lại của danh sách.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 90 | 60 | 50 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 90 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 90 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(5)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 90 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(6)

Vì phần tử có giá trị 60 là phần tử nhỏ nhất kế tiếp nên nó giữ nguyên vị trí, giải thuật tiếp tục với 2 phần tử cuối cùng.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 90 | 70 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 90 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

(7) (8)

Như vậy giải thuật Selection đã sắp xếp xong danh sách sau 8 lần duyệt.

* Độ phức tạp:

Để chọn được phần tử nhỏ nhất, ta cần duyệt qua n phần tử (tốn n-1 phép so sánh) và sau đó hoán vị nó với phần tử đầu tiên của dãy hiện hành. Để tìm phần tử nhỏ tiếp theo, ta cần duyệt qua n-1 phần tử (tốn n-2 phép so sánh). Cứ như vậy, ta thấy ngay thuật toán sẽ tốn:

(n-1) + (n-2) + … + 1= n(n-1)/2 = O(n2)