**Bài 2: Một danh sách các phần tử được lưu trữ trong một danh sách đặc, có các phần tử sau: 40, 70, 20, 60, 90, 10, 50,30. Yêu cầu:**

**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.**

**Bài làm**

0 1 2 3 4 5 6 7

a 40 70 20 60 90 10 50 30

Lần 1: Vị trí đang xét: a [0]

Vị trí chứa giá trị nhỏ nhất từ a[0] đến a[7]: a[6]

# → Đổi chỗ a[6] với a[0]

# ⮆ 10 70 20 60 90 40 50 30

# Lần 2: Vị trí đang xét: a[1]

# Vị trí chứa giá trị nhỏ nhất từ a[1] đến a[7]: a[2]

# → Đổi chỗ a[1] với a[2]

**⮆** 10 20 70 60 90 40 50 30

# Lần 3: Vị trí đang xét: a[2]

# Vị trí chứa giá trị nhỏ nhất từ a[2] đến a[7]: a[7]

# → Đổi chỗ a[2] với a[7]

**⮆** 10 20 30 60 90 40 50 70

# Lần 4: Vị trí đang xét: a[3]

# Vị trí chứa giá trị nhỏ nhất từ a[3] đến a[7]: a[5]

# → Đổi chỗ a[3] với a[5]

**⮆** 10 20 30 40 90 60 50 70

# Lần 5: Vị trí đang xét: a[4]

# Vị trí chứa giá trị nhỏ nhất từ a[4] đến a[7]: a[6]

# → Đổi chỗ a[4] với a[6]

**⮆** 10 20 30 40 50 60 90 70

# Lần 6: Vị trí đang xét: a[5]

# Vị trí chứa giá trị nhỏ nhất từ a[5] đến a[7]: a[5]

# → Không đổi chỗ

**⮆** 10 20 30 40 50 60 90 70

# Lần 7: Vị trí đang xét: a[6]

# Vị trí chứa giá trị nhỏ nhất từ a[6] đến a[7]: a[7]

# → Đổi chỗ a[6] với a[7]

**⮆** 10 20 30 40 50 60 70 90

Mảng sau khi sắp xếp:

**⮆** 10 20 30 40 50 60 70 90

Đánh giá độ phức tạp

TH tốt nhất: - Lần so sánh:

- Lần hoán vị: 0

TH xấu nhất - Lần so sánh:

- Lần hoán vị:

**⮆** Độ phức tạp: O(n2)