**Câu hỏi chương 3**

**Câu 1:**Trong các phương pháp xếp thứ tự đã học, phương pháp nào tối ưu nhất, và kém tối ưu nhất? Tại sao?

* Theo em trong các phương pháp thì Heap Sort tối ưu nhất vì Heap Sort có độ phức tạp về thời gian là O(n log n) cho tất cả các trường hợp (trường hợp tốt nhất, trường hợp trung bình và trường hợp xấu nhất).
* Selection Sort là phương pháp kém tối ưu nhất vì nó có độ phức tạp là O(n2) cho tất cả các trường hợp (trường hợp tốt nhất, trường hợp trung bình và trường hợp xấu nhất).

**Câu 2:** Trong 2 phương pháp tìm kiếm đã học, trường hợp nào thì cả 02 phương pháp đều như nhau? Giải thích tại sao?

* Trong việc xác định giá trị nhỏ nhất/lớn nhất trong một danh sách ( không sắp xếp) thì cả hai phương pháp đều như nhau. Bởi vì thuật toán tìm kiếm tuần tự xuất phát từ đầu hoặc cuối danh sách và lần lượt duyệt qua từng phần tử, chỉ so sánh giá trị cần tìm với giá trị của từng phần tử. Giống như tìm kiếm nhị phân đã được xác sắp xếp thì ta cũng dễ dàng xác định được phần tử nhỏ nhất/lớn nhất trong danh sách này.
* Cả hai phương pháp như nhau khi giá trị cần tìm không nằm trong danh sách duyệt.

**Câu 3:** Ngoài các phương pháp xếp thứ tự đã học, hãy tìm hiểu thêm một phương pháp xếp thứ tự khác, giới thiệu sơ và giải thích.

* Giải thuật Shell Sort:

Shell Sort là một giải thuật sắp xếp một có hiệu quả cao dựa trên giải thuật sắp xếp chèn (Insertion Sort). Giải thuật này tránh các trường hợp phải hoán đổi vị trí của 02 phần tử xa nhau trong Insertion Sort.

Giải thuật này sử dụng giải thuật sắp xếp chọn trên các phần tử có khoảng cách xa nhau, sau đó sắp xếp các phần tử có khoảng cách hẹp hơn. Khoảng cách này được gọi là **khoảng (interval)** – là số vị trí từ phẩn tử này tới phần tử khác.

Interval sẽ nhận các giá trị lần lượt là n/2, n/4, n/8 cho đến khi interval = 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 8 | 2 | 1 | 4 | 7 | 3 | 9 | 6 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

* Cho một dãy như bên dưới, sắp xếp thành dãy tăng.

Với interval = 9/2 = 4, ta chia được cái dãy số với khoảng in=4 lần lượt là:

[5, 4 , 6], [8, 7], [2, 3], [1, 9]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 7 | 2 | 1 | 5 | 8 | 3 | 9 | 6 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Sắp xếp từng dãy con này theo Insertion Sort, ta được dãy:

Với interval = 9/4 = 2, ta chia được cái dãy số với khoảng in=2 lần lượt là:

[4, 2, 5, 3, 6], [7,1,8,9]

Tiếp tục sắp xếp từng dãy con này theo Insertion Sort, ta được dãy:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | 1 | **3** | 7 | **4** | 8 | **5** | 9 | **6** |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Với interval = 9/8 = 1, ta chia được cái dãy số với khoảng in=1 và sau đó sắp xếp theo Insertion Sort ta được dãy:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Giải thuật này khá hiệu quả với các tập dữ liệu có kích cỡ trung bình khi mà độ phức tạp trường hợp xấu nhất và trường hợp trung bình là **O(n)**, với n là số phần tử.