**BÁO CÁO ĐỒ ÁN GIỮA KỲ**

**CÁC THUẬT TOÁN SẮP XẾP**

MSSV: 1512363

Họ và tên: Nguyễn Hoàng Văn Nhã

Môn học: Cấu trúc dữ liệu và giải thuật

TÓM TẮT BÀI BÁO CÁO

I.CÁC THUẬT TOÁN SẮP XẾP

1.Sắp xếp chọn(Selection Sort).

* Ý tưởng.
* Thuật toán.
* Độ phức tạp.
* Đánh giá thuật toán.

2.Sắp xếp chèn(Insertion Sort).

3.Sắp xếp nổi bọt(Bubble Sort).

4.Sắp xếp vun đống(Heap Sort).

5.Sắp xếp trộn(Merge Sort).

6.Sắp xếp nhanh(Quick Sort).

7.Sắp xếp theo cơ số(Radix).

II.KẾT QUẢ THÍ NGHIỆM ĐỐI VỚI TỪNG KIỂU DỮ LIỆU ĐẦU VÀO.

1.Có thứ tự ngược.

* Vẽ đồ thị giữa kích thước giữ liệu và thời gian chạy.
* Nhận xét.

2.Ngẫu nhiên.

3.Gần như có thứ tự.

4.Đã được sắp xếp.

BÁO CÁO

I.CÁC THUẬT TOÁN SẮP XẾP.

1.Sắp xếp chọn(Selection Sort).

* Ý tưởng:

Lần lượt đưa các phần tử cực trị về đầu danh sách.

* Thuật toán:

1. Gán i = 0.
2. Duyệt từ đầu dãy(gán i = 0) đến cuối dãy(n – 1).
3. Tại mỗi phần tử (tại i) ta tìm ta tìm phần tử nhỏ nhất min trong khoảng từ sau phần tử đang duyệt(tức i+1) đến cuối dãy( n -1).
4. Nếu min nhỏ hơn phần tại i thì hoán vị min và phần tử tại i.

* Độ phức tạp.

Do phải thực hiện hai vòng lặp lồng nhau nên độ phức tạp của thuật toán là O(n²).

* Đánh giá thuật toán.

Chi phí của thuật toán không đổi.

Thuật toán có tóc độ thực thi chậm do phải duyệt nhiều lần.

Có thể áp dụng được trong bài toán tìm ra một danh sách đứng đầu (top) gồm ít phần tử.

2.Sắp xếp chèn(Insertion Sort).

* Ý tưởng:

Bắt chước xếp nài của những người chơi bài, muốn sắp xếp một bộ bài người ta phải rút lần lượt từ quân số hai rồi so sánh nó với các quân trước nó và chèn vào vị trí thích hợp.

* Thuật toán:

Xét danh sách A gồm n phần tử được đánh số từ 0 đến n-1.

Duyệt i = 1 (phần tử thứ 2) đến n – 1 (phần tử cuối cùng) với mỗi i ta xét danh sách từ 0

đến i – 1 rồi nếu có vị trí thích hợp thì chèn A[i] vào.

* Độ phức tạp:

Do phải thực hiện hai vòng lặp lồng nhau nên độ phức tạp của thuật toán là O(n²).

* Đánh giá thuật toán:

Độ phức tạp trong trường hợp tốt nhất là O(n) xấu nhất là O(n²)

Phù hợp với sắp xếp các mảng có kích thước nhỏ.

3.Sắp xếp nổi bọt(Bubble Sort).

* Ý tưởng:

Lần lượt so sánh hai phần tử kề nhau nếu nó không nằm đúng vị trí thì đổi chổ cho nhau

* Thuật toán:

Xét danh sách A gồm n phần tử được đánh số từ 0 đến n-1.

Duyệt từ phần tử cuối cùng về phần tử đầu tiên(i = n – 1 về 0).

Ứng với mỗi i ta lần lượt so sánh các cặp gồm hai phần tử kề nhau trong khoảng từ 0 đến i nếu hai phần tử này không đúng vị trí thì đổi chổ cho nhau.

* Độ phức tạp: O(n²)
* Đánh giá thuật toán:

Chi phí của thuật toán không đổi.

Thời gian thực hiện lâu.

4.Sắp xếp vun đống(Heap Sort).

* Ý tưởng:

Dựa vào cấu trúc heap.

* Thuật toán:

Xét danh sách A gồm n phần tử được đánh số từ 0 đến n-1.

1. Hiệu chỉnh A thành một Heap.
2. Đưa phần tử đầu dãy (tức phần tử lớn nhất) về cuối dãy.
3. Lặp lại bước 1 với dãy A giảm đi một phần tử cho đến khi A chỉ còn 1 phần tử thì dừng.

* Độ phức tạp: O(n).
* Đánh giá thuật toán:

Thuật toán phù hợp với việc lấy ra một danh sách các phần tử lớn nhất.

5.Sắp xếp trộn(Merge Sort).

* Ý tưởng:

Chia để trị.

Một danh sách gồm một phần tử thì đã được sắp xếp do đó ta chia nhỏ danh sách ban đầu ra thành đến khi nào tạo ra các sanh sách con chỉ gồm một phần tử rồi trộn các danh sách con đó lại.

* Thuật toán:

Nếu dãy gồm 1 hoặc 0 phần tử thì dãy đã được sắp xếp.

Ngược lại thì

1. Chia dãy ra làm hai dãy con.
2. Dùng Merge Sort để sắp xếp trên hai dãy con đó.
3. Trộn hai dãy con đã được sắp xếp thành một dãy đã được sắp xếp.

* Độ phức tạp: O(n).
* Đánh giá thuật toán:

Số lần chia dãy con là .

Chi phí của thuật toán không đổi.

6.Sắp xếp nhanh(Quick Sort).

* Ý tưởng:

Chọn một phần tử chốt và đưa các phần tử nhỏ hơn chốt về một phía các phần tử lớn hơn chốt về một phía.

* Thuật toán:

1. Nếu mảng có một phần tử thì dừng lại.
2. Chọn phần tử chốt.
3. Đưa các phần tử lớn hơn chốt về một phía và các phần tử nhỏ hơn về phía đối diện
4. Thực hiện sắp xếp nhanh trên các phía của phần tử chốt.

* Độ phức tạp: O(n).
* Đánh giá thuật toán:

Độ phức tạp của thuật toán phụ thuộc vào phần tử chốt.

7.Sắp xếp theo cơ số(Radix).

* Ý tưởng:

Dựa vào các cơ số của từng phần tử trong mảng để sắp xếp.

* Thuật toán:

1. Tìm phần tử lớn nhất trong mảng.
2. Duyệt từ cơ số nhỏ nhất đến cơ số lớn nhất của phần tử lớn nhất.
3. Với mỗi cơ số ta sắp các phần tử trong mảng vào các “thùng” theo cơ số đang xét. Rồi lấy ra tùy theo yêu cầu.

* Độ phức tạp: O(n).
* Đánh giá thuật toán:

Thuật toán không dựa vào việc so sánh các phần tử.

Tốn bộ nhớ.

II.KẾT QUẢ THÍ NGHIỆM ĐỐI VỚI TỪNG KIỂU DỮ LIỆU ĐẦU VÀO.

1.Có thứ tự ngược.

Nhận xét:

* Selection sort và Bubble sort chạy chậm nhất do nó không phụ thuột vào dữ liệu vào.
* Quick sort chạy nhanh nhất do dữ liệu vào đã có thứ tự ngược.
* Insertion sort chạy khá nhanh do việc tìm phần tử cực trị khá dễ dàng.
* Radix sort và Merge sort không bị ảnh hưởng bởi thứ tự của dữ liệu vào.
* Heap sort chạy nhanh vì đầu vào có thứ tự ngược.

2.Ngẫu nhiên.

Nhận xét:

* Selection sort và Bubble sort chạy chậm nhất do nó không phụ thuột vào dữ liệu vào.
* Quick sort và Radix sort không bị ảnh hưởng bởi dữ liệu vào có thứ tự hay chưa.
* Insertion sort chạy chậm do việc tìm phần tử cực trị không còn dễ dàng nữa.
* Radix sort và Merge sort không bị ảnh hưởng bởi thứ tự của dữ liệu vào.

3.Gần như có thứ tự.

Nhận xét:

* Selection sort và Bubble sort chạy chậm nhất do nó không phụ thuột vào dữ liệu vào.
* Quick sort chạy nhanh nhất do dữ liệu vào đã gần như có thứ tự.
* Radix sort và Merge sort không bị ảnh hưởng bởi thứ tự của dữ liệu vào.
* Heap sort chạy nhanh vì đầu vào có thứ tự.

4.Đã được sắp xếp.

Nhận xét:

* Selection sort và Bubble sort chạy chậm nhất do nó không phụ thuột vào dữ liệu vào.
* Quick sort chạy nhanh nhất do dữ liệu vào đã có thứ tự .
* Radix sort và Merge sort không bị ảnh hưởng bởi thứ tự của dữ liệu vào.
* Heap sort chạy nhanh vì đầu vào có thứ tự.