

Sắp xếp trên mảng

Sắp xếp cơ bản

Selection sort
Insertion sort
Bubble sort

Sắp xếp nâng cao

Quick sort
Merge sort
Heap sort

Problem: Cho mảng A có n phần tử. Hãy sắp xếp dãy A theo thứ tự tăng dần (hoặc giảm dần)

Các thuật toán sắp xếp dựa trên phép so sánh

Name	Average	Worst	Memory	Stable	Method
Bubble sort	—	$O(n^2)$	$O(1)$	Yes	Exchanging
Cocktail sort	—	$O(n^2)$	$O(1)$	Yes	Exchanging
Comb sort	$O(n \log n)$	$O(n \log n)$	$O(1)$	No	Exchanging
Gnome sort	—	$O(n^2)$	$O(1)$	Yes	Exchanging
Selection sort	$O(n^2)$	$O(n^2)$	$O(1)$	No	Selection
Insertion sort	$O(n + d)$	$O(n^2)$	$O(1)$	Yes	Insertion
Shell sort	—	$O(n \log^2 n)$	$O(1)$	No	Insertion
Binary tree sort	$O(n \log n)$	$O(n \log n)$	$O(n)$	Yes	Insertion
Library sort	$O(n \log n)$	$O(n^2)$	$O(n)$	Yes	Insertion
Merge sort	$O(n \log n)$	$O(n \log n)$	$O(n)$	Yes	Merging
In-place merge sort	$O(n \log n)$	$O(n \log n)$	$O(1)$	Yes	Merging
Heapsort	$O(n \log n)$	$O(n \log n)$	$O(1)$	No	Selection
Smoothsort	—	$O(n \log n)$	$O(1)$	No	Selection
Quicksort	$O(n \log n)$	$O(n^2)$	$O(\log n)$	No	Partitioning
Introsort	$O(n \log n)$	$O(n \log n)$	$O(\log n)$	No	Hybrid
Patience sorting	—	$O(n^2)$	$O(n)$	No	Insertion

So sánh các thuật toán sắp xếp

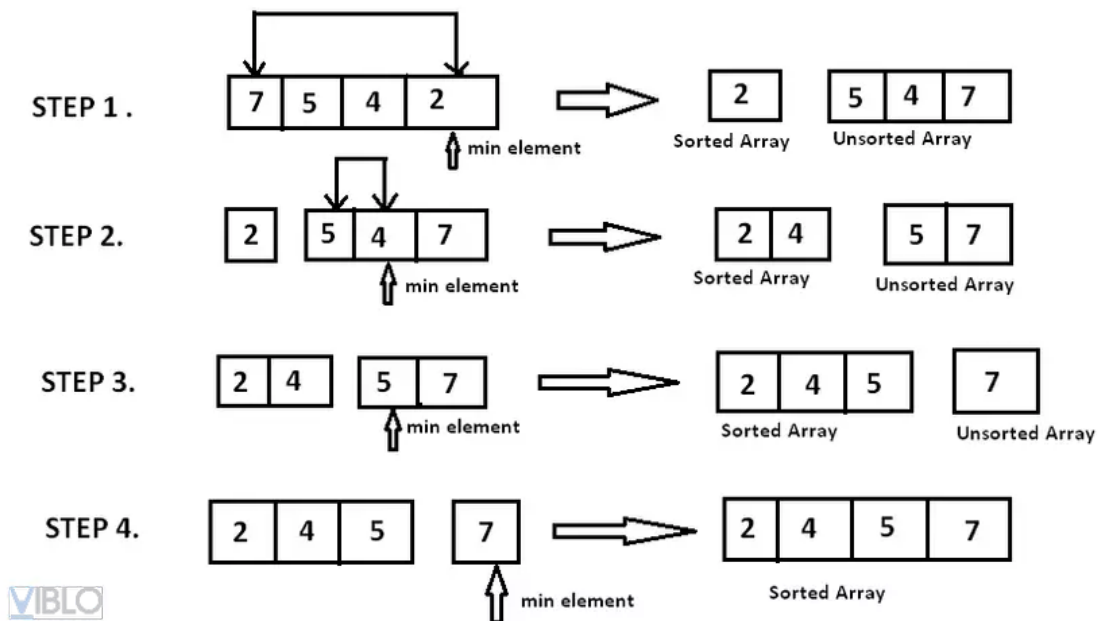
Link visualize một số giải thuật sắp xếp: <https://visualgo.net/en/sorting>

Sắp xếp cơ bản

Selection sort

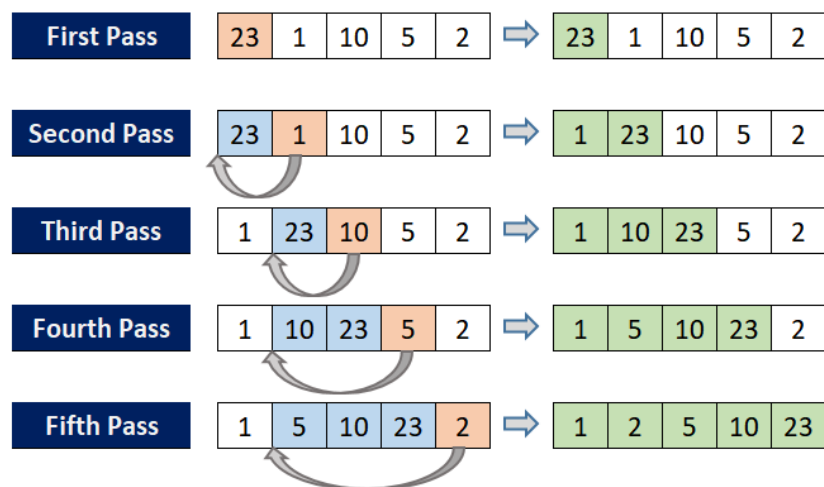
- Chiến thuật: Chia A thành 2 phần: chưa sắp xếp bên phải và đã sắp xếp bên trái.
→ Chọn số **nhỏ nhất** trong dãy chưa sắp xếp → **đổi chỗ** lên đầu dãy chưa sắp xếp (hay là cuối dãy đã sắp xếp)

→ Lặp lại hành động tìm số nhỏ nhất trong dãy chưa sắp xếp và đổi lên đầu cho tới hết dãy chưa sắp xếp



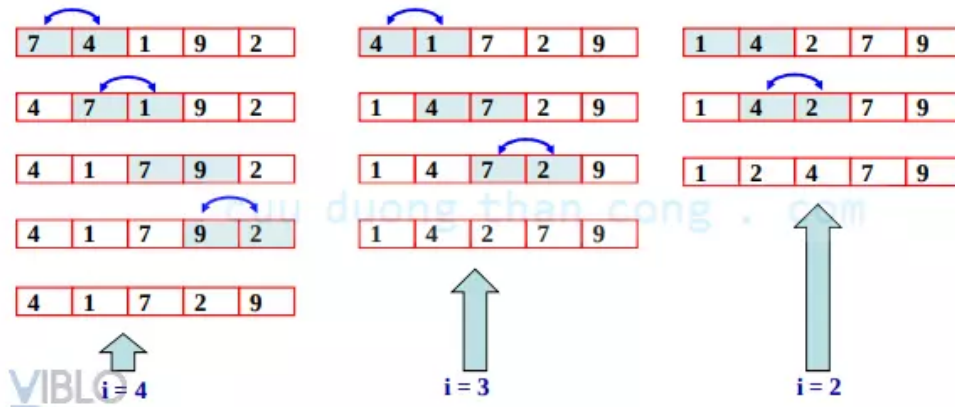
Insertion sort

- Chiến thuật: Chia dãy A thành 2 phần: Phần đã sắp xếp (bên trái) và chưa sắp xếp (bên phải)
 - Duyệt **từng phần tử** (khác Selection sort là lấy phần tử min) và đưa nó vào đúng vị trí trong dãy sắp xếp (tức chèn thêm phần tử đó vào dãy đã sắp xếp sao cho sau khi chèn dãy vẫn được sắp xếp)
 - Lặp lại cho tới khi hết phần tử trong dãy chưa sắp xếp
- Ưu điểm: Nếu danh sách đã gần đúng thứ tự, Insertion Sort sẽ chạy rất nhanh (Ví dụ bạn cần sắp xếp Highscore trong game)



Bubble sort

- Chiến thuật: Xét lần lượt các cặp 2 phần tử liên tiếp
 - Nếu phần tử đứng sau nhỏ hơn phần tử đứng trước, ta đổi chỗ 2 phần tử. Nói cách khác, phần tử **nhỏ nhất sẽ nổi** lên trên
 - Lặp lại đến khi không còn 2 phần tử nào thỏa mãn

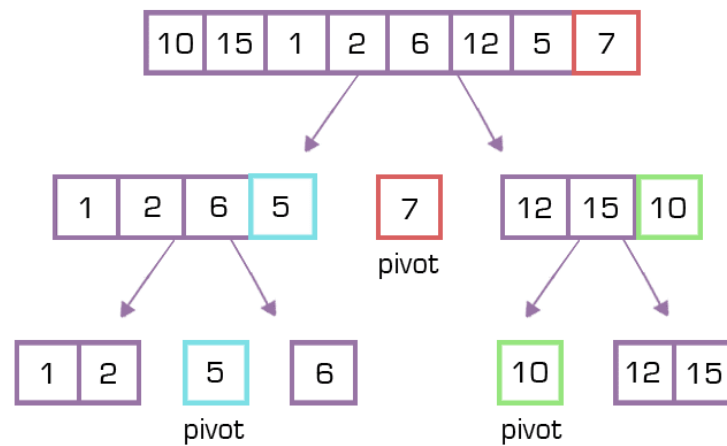


- Ưu điểm: Code đơn giản dễ hiểu, không tốn thêm bộ nhớ

Sắp xếp nâng cao

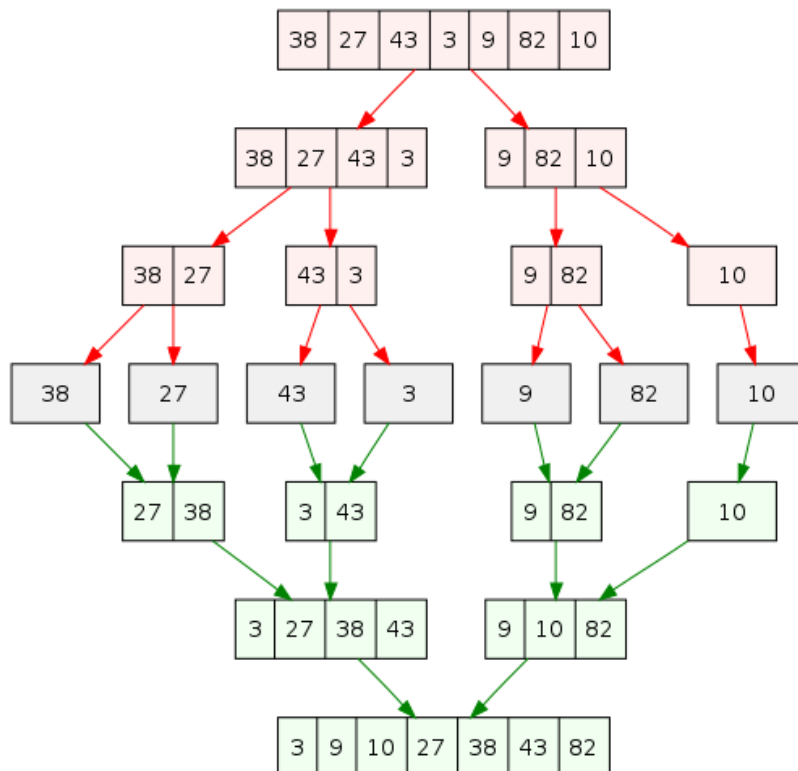
Quick sort

- Chiến thuật: Chia để trị (hoạt động theo kiểu đệ quy)
 - Chia dãy thành 2 mảng con, một phần “nhỏ” ở trước và một phần “lớn” ở sau → **base case**: dãy có 1 phần tử
 - Chọn 1 khóa **pivot** là phần tử bất kì
 - Mọi phần tử nhỏ hơn pivot → Đẩy lên trước; phần tử lớn hơn pivot → đẩy xuống sau
 - Chiến thuật tương tự với từng mảng con (gọi đệ quy) → tới khi quy về base case



Merge sort

- Chiến thuật: Chia để trị (hoạt động theo kiểu đệ quy)
 - Chia dãy thành 2 phần A, B và sắp xếp từng phần → **base case**: dãy có 1 phần tử
 - Gộp (merge) 2 phần A và B lại với nhau bằng cách:
 - Tạo dãy X mới chứa các phần tử đã sắp xếp
 - So sánh 2 phần tử đầu tiên của 2 phần A và B → phần tử nào nhỏ hơn ta chuyển vào X (xóa tương ứng trong A, B)
 - Tiếp tục làm vậy cho tới khi cho tất cả phần tử từ A, B vào X



Heap sort

- Chiến thuật: **Lưu mảng vào cấu trúc dữ liệu Heap**

▼ Về Heap

Heap là cây nhị phân gần hoàn chỉnh có 2 tính chất

- Tính cấu trúc: tất cả các mức đều đầy đủ các node con (ngoại trừ mức cuối cùng (điền từ trái sang phải))
- Tính có thứ tự: Với mỗi nút x , có $Parent(x) \geq x$

2 loại Heap → Max heap (phần tử lớn nhất ở gốc) và Min heap (phần tử nhỏ nhất ở gốc)

→ Ở mỗi bước lấy ra phần tử **lớn nhất trong max-heap** (chính là gốc) và cho vào mảng đã sắp xếp rồi đưa node con cuối cùng lên làm gốc → tiến hành vun lại đống (khôi phục tính chất max heap) bằng cách:

- Giả sử có nút i có giá trị bé hơn con của nó và cây con trái và phải của i đều là max heap
 - Đổi chỗ i với con lớn hơn đó
 - Di chuyển xuống theo cây
 - Tiếp tục quá trình cho tới khi nút i không còn bé hơn con

→ Tiếp tục quá trình tới khi hết heap

10	4	8	5	12	2	6	11	3	9	7	1
----	---	---	---	----	---	---	----	---	---	---	---