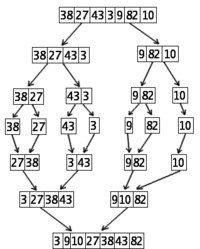
* **Lùa bò vào chuồng:** file lo trinh
* **Duyệt mảng:** file lo trinh
* Duyệt xuôi, duyệt ngược : duyệt 1->n, n->1
* Sử dụng biến “lính canh” : tìm max,min
* Cộng dồn: <https://www.youtube.com/watch?v=F68H8ESSOIs&ab_channel=thisisbac>
  + Loại 1:
    - Bth: tính tg từ 1-i thì for 1-i ( nhiều lần)
    - Prefix sum: in f[i] thôi
  + Loại 2 (L/R/V) :
    - Bth: for L-R
    - Prefix sum: cập nhập L và R sau đó áp dụng loại 1
  + Loại 3 (mảng 2 chiều):
* Kỹ thuật ô ăn quan: xem file ve PP duyet
* Kỹ thuật đánh dấu: sàng số nguyên tố, xem file PP duyet
* **Sắp xếp:**
  + <https://topdev.vn/blog/thuat-toan-sap-xep-trong-c/#:~:text=M%E1%BB%A5c%20%C4%91%C3%ADch%20c%E1%BB%A7a%20vi%E1%BB%87c%20s%E1%BA%AFp,v%E1%BB%81%20b%C3%A0i%20to%C3%A1n%20t%C3%ACm%20ki%E1%BA%BFm>.
  + <https://viblo.asia/s/cac-giai-thuat-sap-xep-VgZvEveMKAw>
  + Sắp xếp nổi bọt (Bubble Sort): sắp xếp bth, đơn giản nhất, chậm nhất
  + Sắp xếp chọn (Selection Sort): vừa sắp xếp vừa tìm min max
  + Sắp xếp chèn (Insertion Sort): xếp giống cách xếp bài,thích hợp đối với mảng đã được sắp xếp một phần hoặc mảng có kích thước nhỏ
  + Sắp xếp trộn (Merge Sort): tách mảng làm 2 liên tục, rồi gộp lại theo thứ tự   
      
    - Thuật toán trộn tự nhiên khác thuật toán trộn trực tiếp ở chỗ thay vì luôn cứng nhắc phân hoạch theo dãy con có chiều dài k, việc phân hoạch sẽ theo đơn vị là đường chạy. ta chỉ cần biết số đường chạy của a sau lần phân hoạch cuối cùng là có thể biết thời điểm dừng của thuật toán vì dãy đã có thứ tự là dãy chi có một đường chạy.

- Một nhược điểm lớn nữa của thuật toán trộn là khi cài đặt thuật toán đòi hỏi thêm không gian bộ nhớ để lưu các dãy phụ b, c. Hạn chế này khó chấp nhận trong thực tế vì các dãy cần sắp xếp thường có kích thước lớn. Vì vậy thuật toán trộn thường được dùng để sắp xếp các cấu trúc dữ liệu khác phù hợp hơn như danh sách liên kết hoặc file.

* + Sắp xếp nhanh (Quick Sort): Tốc độ sắp xếp nhanh, Được sử dụng trong nhiều thư viện của các ngôn ngữ như C++, Java, ..., Phụ thuộc vào cách chọn phần tử chốt. Không ổn định.
* **Danh sách liên kết**
  + <https://topdev.vn/blog/danh-sach-lien-ket-don-trong-c/>
  + <https://topdev.vn/blog/danh-sach-lien-ket-don-va-cac-bien-the-singly-linked-list-and-its-variants/>
  + Lợi ích chính của danh sách liên kết so với mảng thông thường là các phần tử danh sách có thể được chèn hay xóa một cách dễ dàng mà không cần phân bổ lại hoặc sắp xếp lại toàn bộ cấu trúc vì các mục dữ liệu không cần được lưu trữ liên tục trong bộ nhớ hay trên đĩa
* **Số học :** <https://www.youtube.com/watch?v=dPoIBEl0sSY&ab_channel=SOnline>
* **Xử lí xâu, chuỗi:** 
  + <https://viblo.asia/p/lap-trinh-c-co-ban-bai-8-phan-2-xau-ki-tu-cac-thao-tac-xu-ly-tren-xau-bWrZnAn9Kxw#_2-chuan-hoa-xau-12>
  + <https://isinhvien.com/tong-hop-cac-thao-tac-xu-li-chuoi-string-trong-c/>
* **Tìm kiếm nhị phân**
  + <https://gochocit.com/ky-thuat-lap-trinh/thuat-toan-tim-kiem-nhi-phan-binary-search>
  + Thuật toán Binary Serach (Tìm kiếm nhị phân) là một thuật toán tìm kiếm tuyến tính cao cấp hơn với thời gian chạy là O(logN).Đối với các danh sách lớn, thuật toán này tốt hơn hẳn tìm kiếm tuyến tính, nhưng nó đòi hỏi danh sách phải được sắp xếp từ trước và đòi hỏi khả năng truy nhập ngẫu nhiên (random access).Thuật toán tìm kiếm nhị phân thường dùng để tìm kiếm phần tử trong một danh sách đã được sắp xếp, ví dụ như trong một danh bạ điện thoại sắp xếp theo tên, có thể tìm kiếm số điện thoại của một người theo tên người đó.Thuật toán tìm kiếm nhị phân chạy nhanh hơn tìm kiếm tuần tự nhưng cũng có một số nhược điểm. Nó chậm hơn bảng băm.Nếu nội dung danh sách bị thay đổi thì danh sách phải được sắp xếp lại trước khi sử dụng tìm kiếm nhị phân.Mà thao tác này thường tốn nhiều thời gian.