# Chia để trị

## Tìm kiếm nhị phân(xong) - O(log2n)

## Tìm GT min max(xong) - 0(n)

## Merge sort(xong) - O(nlogn)

## Quick sort - O(nlogn) trường hợp trung bình

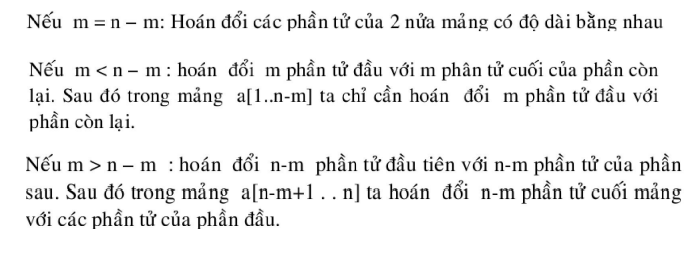
## Dãy con lớn nhất(xong) - O(nlogn)

dựa trên MSV

## Tính lũy thừa (xong) -O(logn)

dựa trên n chẵn hay lẻ

## Hoán đổi vị trí phần tử mảng (xong)



# Tham lam

Duyệt vét cạn, trong tương lai ko xem xét lại quyết định trong quá khứ, không cho kết quả tối ưu.

## Cái túi (xong) - O(n)

## Người đi du lịch (xong) - O(n^2)

## Dijkstra-đường đi ngắn nhất – O(n^2) (xong)

tối ưu

Tìm đỉnh V/S có nhãn nhỏ nhất kề với S để đưa vào S

## Cây bao trùm nhỏ nhất (xong) - O(n^3)

prim và kruskal (cây khung nhỏ nhất)

Số cạnh = số đỉnh -1

Prim: đỉnh đã được xét

Kruscal thì lấy cạnh nhỏ nhất mặc dù đỉnh chưa dc chọn miễn ko tạo thành chu trình (dựa trên mỗi đỉnh 1 màu, khi nối được với nhau thì đỉnh sau mang màu của đỉnh trước)

## **Bài toán tô màu (xong)** - O(n^2)

2 đỉnh kề nhau có màu khác nhau

## Các khoảng không giao nhau (xong)

-sx theo thứ tự tăng dần đầu mút trái

-sx theo thời gian thực hiện công việc

-sx theo thứ tự tăng dần đầu mút phải

# Quy hoạch động-tối ưu – tốc độ nhanh- nhược điểm: không dễ tìm ra lời giải

## Fibonacy - O(n)

Chia để trị: tính bài toán con nhiều lần, theo hướng đệ quy, từ trên xuống dưới

Quy hoạc động: từ dưới lên trên, tính trước lời giải bài toán con, lưu vào bộ nhớ

## Cái túi (khó chắc ko hỏi thêm lý thuyết) - O(n\*b)

b là trọng lượng túi

## Dãy con có tổng lớn nhất- O(n)

## Xâu con chung dài nhất- O(n^2)

## Đường đi ngắn nhất (floyd)- O(n^3)

Giữa 2 cặp đỉnh bất kỳ

# Quay lui

## 8 hậu (xong) O(n!)

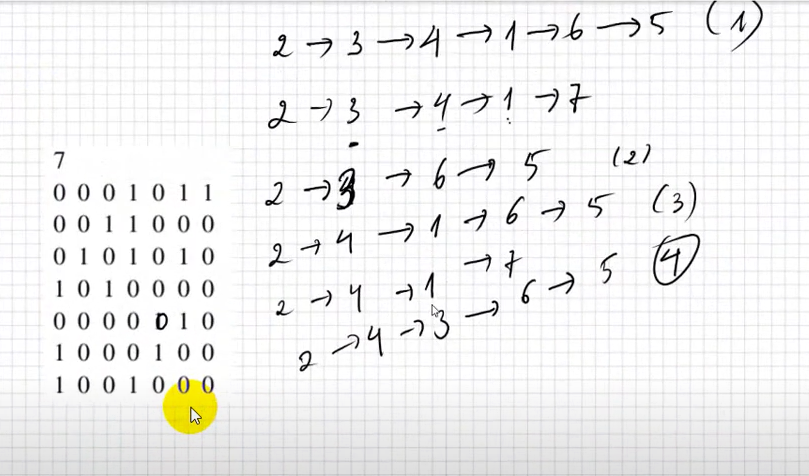
## Mã đi tuần (xong)- O(8^n bình)

## Liệt kê hoán vị

## Liệt kê nhị phân

## Duyệt đồ thị- DFS- O(n^n)

Last in first out-đệ quy quay lui



## Người đi du lịch (liệt kê hoán vị)- O(n^n)

## Cái túi (liệt kê nhị phân)- O(n)

# Nhánh cận

Cải tiến từ phương pháp quay lui, tìm lời giải tối ưu của bài toán

## Người đi du lịch

## Cái túi