

# Giải thuật tham lam Greedy algorithm

Nguyễn Văn Huy

[huyite.vn@gmail.com](mailto:huyite.vn@gmail.com)

Ngày 13 tháng 3 năm 2024

- 1 Giới thiệu
- 2 Một số bài toán minh họa
  - Bài toán cái túi
  - Bài toán đổi tiền
  - Bài toán TSP

# Bài toán tối ưu tổ hợp

## Định nghĩa

**Bài toán tối ưu tổ hợp** là bài toán tìm phương án tối ưu trên tập các cấu hình tổ hợp. Nghiệm của bài toán cũng là một vector  $x$  gồm  $n$  thành phần sao cho:

- $x = (x_1, x_2, \dots, x_n)$
- $x \in D_n$ ,  $D_n$  là tập hữu hạn.
- $f(x) \rightarrow \min/\max$ .

# Ý tưởng

Tìm nghiệm nghiệm chấp nhận được trong tập xác định. Trong mỗi bước lặp ta chọn một nghiệm tốt hơn, đến khi không chọn được phương án tiếp theo thuật toán dừng.

## Ưu điểm

- Thời gian thực thi nhanh.
- Độ phức tạp thuật toán đa thức.

## Nhược điểm

- Chỉ tìm được phương án tốt, chưa chắc tối ưu.

## Bài toán Knapsack

Có  $n$  đồ vật, mỗi đồ vật có giá trị  $v_i$ , trọng lượng  $w_i$ . Một người có khả năng mang được trọng lượng  $W$ , hãy lựa chọn các đồ vật sao cho tổng trọng lượng các đồ vật nhỏ hơn  $W$  và tổng giá trị các đồ vật lớn nhất có thể.

## Math Model

- Biến quyết định

$$x_i = \begin{cases} 1 & \text{Chọn đồ vật } i \\ 0 & \text{otherwise} \end{cases}$$

- Tham số:  $v_i$  giá trị đồ vật thứ  $i$ ,  $w_i$  trọng lượng đồ vật thứ  $i$
- Ràng buộc

$$\sum_{i=1}^n w_i \times x_i \leq W \quad (1)$$

- Hàm mục tiêu

$$f(x_{i,j}) = \sum_{i=1}^n v_i \times x_i \longrightarrow \max \quad (2)$$

# Greedy algorithm

---

## Algorithm 1 Greedy

---

```
1: procedure KNAPSACK( $x_i$ )
2:   Sort( $X$ )
3:   for  $i = 1$  to  $n$  do
4:     if  $\sum w_j$  được chọn +  $w_i \leq W$  then
5:       chọn đồ thứ  $i$ 
```

---

## Ví dụ

Trọng lượng tối đa 27 và có 4 đồ vật như bảng dưới

Bảng 1: Knapsack

Loại đồ vật	Trọng lượng	Giá trị
A	15	30
B	10	25
C	2	2
D	4	6

A, B, C



## Ví dụ

Trọng lượng tối đa 27 và có 4 đồ vật như bảng dưới

Bảng 2: Knapsack

Loại đồ vật	Trọng lượng	Giá trị
A	15	30
B	10	25
C	1	4
D	2	6
E	1	4

A, B, D but not best

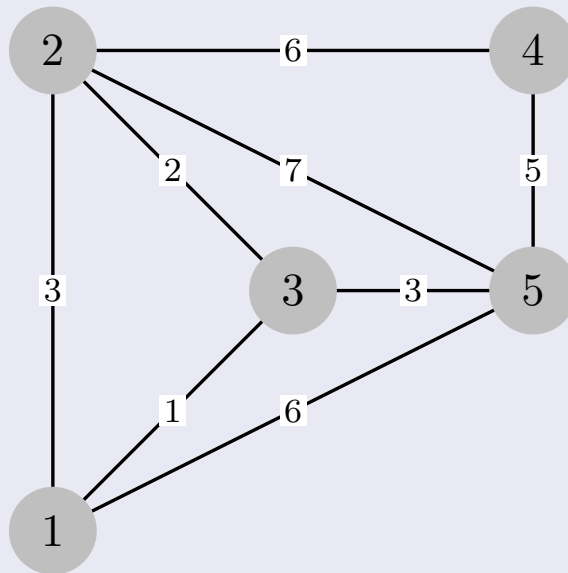
A, B, C, E -> best solution

# Bài toán đổi tiền

Trong máy có chuẩn bị sẵn các loại tiền  $10K$ ,  $20K$ ,  $50K$  và  $100K$ . Giả sử số lượng không hạn chế. Khi có một khách hàng cần rút  $N$  đồng, với  $N$  chia hết cho  $10K$ . Tìm một phương án trả  $N$  đồng và số lượng tờ ít nhất.

Có  $n$  thành phố,  $d[i, j]$  là chi phí để di chuyển từ thành phố  $i$  đến thành phố  $j$ . (Nếu không có đường đi thì  $d[i, j] = \infty$ ). Một người muốn đi du lịch qua tất cả các thành phố, mỗi thành phố một lần rồi trở về nơi xuất phát sao cho tổng chi phí là nhỏ nhất. Hãy xác định một đường đi như vậy.

## Ví dụ



## Mark and Toys

Mark and Jane are very happy after having their first kid. Their son is very fond of toys, so Mark wants to buy some. There are  $N$  different toys lying in front of him, tagged with their prices, but he has only  $K$  \$. He wants to maximize the number of toys he buys with this money. Now, you are Mark's best friend and have to help him buy as many toys as possible.

sample

7 50

1 12 5 111 200 1000 10

# Exercise

## Max Min

Given a list of  $N$  integers, your task is to select  $K$  integers from the list such that its unfairness is minimized.

if  $(x_1, x_2, \dots, x_k)$  are  $K$  numbers selected from the list, the unfairness is defined as  $\max(x_1, x_2, \dots, x_k) - \min(x_1, x_2, \dots, x_k)$  where  $\max$  denotes the largest integer among the elements of  $K$ , and  $\min$  denotes the smallest integer among the elements of  $K$ .

## Example:

7 3

10 100 300 200 1000 20 30

$$\max(10, 20, 30) - \min(10, 20, 30) = 30 - 10 = 20$$

## Chia tài sản

Trong ngày sinh nhật, hai anh em Tom và Alice nhận được  $N$  ( $N < 10$ ) đồ chơi. Trên đồ chơi  $i$  có ghi giá tiền  $X_i$ . Hai anh em quyết định mỗi người phải có trách nhiệm bảo quản một phần số quà và phân chia sao cho chênh lệch tổng giá trị tiền đồ chơi mà mỗi người bảo quản là ít nhất. Hãy giúp Tom và Alice phân chia. Viết cho phép người dùng nhập  $N$  đồ vật và giá trị các đồ vật  $X_i$ .

- Travelling Salesman Problem
- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm
- Dijkstra's Minimal Spanning Tree Algorithm
- Graph - Map Coloring
- Graph - Vertex Cover
- Knapsack Problem
- Job Scheduling Problem