ĐẠI HỌC ĐÔNG Á
Tạo dựng con đường thành công

GIẢI THUẬT NÂNG CAO

Chương 3:
Các hướng tiếp cận thiết kế giải thuật
(Giải thuật Tham lam - Greedy Algorithm)

ĐẠI HỌC ĐÔNG Á
Tạo dựng con đường thành công

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

MỤC TIÊU BÀI HỌC

- **Kiến thức**
  - ✓ Hiểu được khái niệm và tư tưởng cốt lõi của Giải thuật Tham lam - (Greedy Algorithm)
  - ✓ Biết cách thiết kế giải thuật tham lam cho một số bài toán cụ thể
  - ✓ So sánh giải thuật tham lam với các giải khác: Chia để Trị, Qui hoạch động
- **Kỹ năng**
  - ✓ Vận dụng kiến thức để giải một số bài toán kinh điển bằng giải thuật tham lam

ĐẠI HỌC ĐÔNG Á
Tạo dựng con đường thành công

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Mở đầu

❑ **Vấn đề tìm kiếm giải pháp tối ưu:**

- Chia bài toán thành nhiều bài toán c
- Giải quyết các bài toán con
- Giải pháp của các bài toán con sẽ là giải ph

Có giải thuật nào độ phức tạp không cao (có tính khả thi với không gian dữ liệu số lớn) những kết quả vẫn chấp nhận được (gần đúng) ?

Chia dể Trị (Divide & Conquer)	Qui hoạch động (Dynamic Programming)
✓ Giải quyết **tất cả các bài toán con** ✓ Độ phức tạp cao (thường hàm số mũ hoặc giai thừa) ✓ Dễ thiết kế, cài đặt (thường là đệ qui)	✓ **Ghi nhớ lại các giải pháp của các bài toán con** để tránh các xử lý trùng lặp. ✓ Độ phức tạp thấp hơn (thường hàm đa thức) ✓ Khó thiết kế và cài đặt giải pháp

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Khái niệm giải thuật Tham lam (Greedy Algorithm)

❑ Khái niệm

Là chiến lược giải thuật chọn phương án tốt nhất tại mỗi bước với hy vọng rằng tổng thể là tối ưu.

❑ Nguyên tắc

- Chọn lựa cục bộ tối ưu → hướng tới tối ưu toàn cục
- Chỉ giải quyết một bài toán con (không xét tất cả các bài toán con)
- Xây dựng giải pháp từng bước một
  - Ở mỗi bước, thực hiện chọn lựa tốt nhất tại thời điểm đó (tối ưu cục bộ)
  - Không quay lại xem xét các quyết định đã lựa chọn
  - Hy vọng kết quả thu được là giải pháp tối ưu (toàn cục)

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Ưu và nhược điểm của Greedy Algorithm

- **Ưu điểm**
  - Dễ thiết kế
  - Dễ cài đặt
  - Độ phức tạp thấp
- **Nhược điểm**
  - Không phải luôn cho giải pháp tối ưu (thường gần tối ưu)
  - Khó để chứng minh thuật toán cho giải pháp tối ưu

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Cấu trúc tổng quát

```
thamlam(C: tập hợp các ứng cử viên)
// hàm trả về giải pháp tối ưu, gồm các ứng cử viên
begin
    S = ∅          // S là giải pháp tối ưu
    while (C ≠ ∅ và S chưa là giải pháp) do
        x = chọn(C) // chọn x từ tập C theo tiêu chí của hàm chọn
        C = C - {x}
        if (S U {x} có triển vọng là giải pháp) then S:=S U {x}
        endif
    endwhile
        if (S là lời giải) then return S
        else return 0
        endif
end
```

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số bài toán áp dụng Greedy Algorithm điển hình

Bài toán 1: Balo phân số (Fractional Knapsack)

Cho n món đồ, mỗi món có trọng lượng w[i] và giá trị v[i]. Chọn sao cho tổng trọng lượng ≤ W, giá trị lớn nhất. Cho phép lấy 1 phần món đồ.

- **Chiến lược tham lam**:
  - Tính value/weight cho từng món.
  - Sắp xếp giảm dần theo tỷ lệ đó.
  - Lấy từ trên xuống đến khi đầy balo.

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số bài toán áp dụng Greedy Algorithm điển hình

Bài toán 2: Hoạt động không giao nhau (Activity Selection)

Mỗi hoạt động có thời gian bắt đầu và kết thúc. Chọn nhiều nhất số hoạt động không trùng thời gian.

- **Chiến lược tham lam**:
  - Sắp xếp theo thời gian kết thúc tăng dần.
  - Luôn chọn hoạt động kết thúc sớm nhất không trùng với hoạt động trước đó.

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số bài toán áp dụng Greedy Algorithm điển hình

Bài toán 3: Huffman Coding

Cho tần suất xuất hiện các ký tự, xây dựng mã nhị phân sao cho tổng chiều dài chuỗi mã hóa là nhỏ nhất.

- **Chiến lược tham lam**:
  - Gộp 2 ký tự/tập có tần suất nhỏ nhất lại thành nút cha → lặp lại.

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

- Bài toán
  - Có một chiếc xe ôtô duy nhất và có nhiều khách hàng yêu cầu được thuê xe. Mỗi yêu cầu thuê xe có một thời điểm bắt đầu thuê và một thời điểm kết thúc thuê.
  - Vấn đề: sắp xếp việc cho thuê làm sao để *số khách hàng* được thuê xe là nhiều nhất

- Hình thức hoá
  - Giả sử S = $\{a_1, a_2, ..., a_n\}$ tập hợp các yêu cầu thuê xe
  - Mỗi $a_i \in S$, $s(a_i)$ là thời điểm bắt đầu thuê và $f(a_i)$ là thời điểm kết thúc thuê
  - Gọi F là tập hợp lớn nhất các yêu cầu thoả mãn:
    - Hai yêu cầu bất kỳ phải lệnh nhau về thời gian, nghĩa là yêu cầu tiếp theo chỉ được bắt đầu khi yêu cầu trước đó kết thúc
    - Hay: $\forall a_1 \in S, a_2 \in S: s(a_1) \leq s(a_2) \Rightarrow f(a_1) \leq f(s_2)$

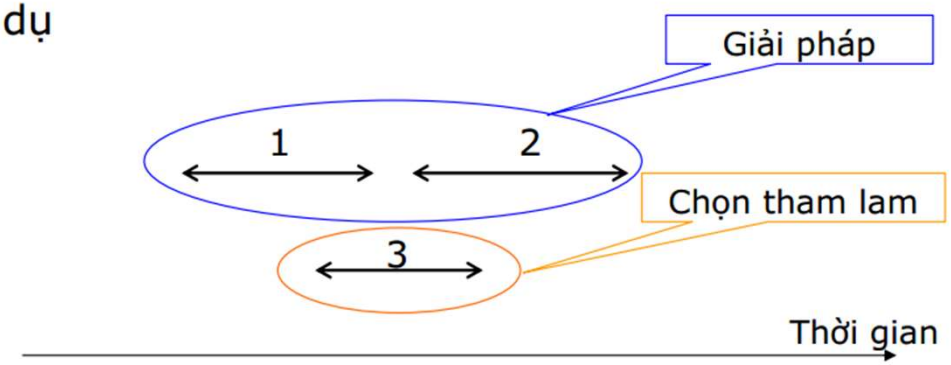
GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

❑ **Phép thử 1**

- Sắp xếp các yêu cầu thuê xe theo thứ tự tăng dần thời gian thuê
- Chọn tham lam
  - chọn yêu cầu có thời gian thuê ngắn nhất
- Ví dụ



⇨ Thuật toán không cho giải pháp tối ưu

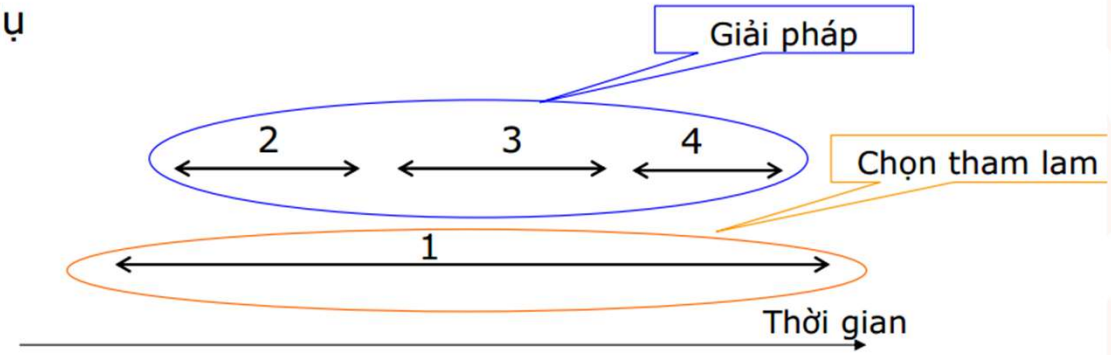
GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

❑ **Phép thử 2**

- Sắp xếp các yêu cầu thuê xe theo thứ tự thời điểm bắt đầu thuê
- Chọn tham lam
  - chọn yêu cầu có thời điểm bắt đầu thuê sớm nhất
- Ví dụ



Thuật toán cũng không cho giải pháp tối ưu

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

❑ **Phép thử 3**

- Sắp xếp các yêu cầu theo thứ tự thời điểm kết thúc thuê tăng dần
- Chọn tham lam
  - chọn yêu cầu có thời điểm kết thúc thuê sớm nhất
- Thuật toán cho giải pháp tối ưu
  - Tại sao ?
  - Chứng minh !

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

Chứng minh tối ưu của thuật toán (phép thử 3)

- Giả sử $F=\{x_1, x_2, \ldots, x_p\}$ là giải pháp đạt được bởi thuật toán tham lam và $G=\{y_1, y_2, \ldots, y_q\}$ với $q \geq p$ là một giải pháp tối ưu (cho phép thực hiện nhiều yêu cầu nhất)
- Cần chứng minh F là giải pháp tối ưu, nghĩa là p=q

- Giả sử các phần tử của các tập hợp F và G được sắp xếp theo thứ tự thời điểm kết thúc thuê tăng dần
- Nếu G không chứa F, thì phải tồn tại k sao cho: $\forall i<k$, $x_i=y_i$ và $x_k \neq y_k$. x_k được chọn bởi thuật toán tham lam, nên x_k có $f(x_k)$ nhỏ nhất và thời điểm bắt đầu sau $x_{k-1}=y_{k-1}$. Vậy $f(x_k) \leq f(y_k)$. Thay G bởi $G'=\{y_1, y_2, \ldots, y_{k-1}, x_k, y_{k+1}, \ldots, y_q\}$ thoả mãn ràng buộc sự lệnh nhau về thời gian của các yêu cầu. Vậy G' cũng là một giải pháp tối ưu mà có số yêu cầu trùng với F nhiều hơn so với G.
- Lặp lại bước trên, cuối cùng có được G'' chứa F mà $|G''|=|G|$
- Nếu G'' có chứa yêu cầu không thuộc F (tức là các yêu cầu bắt đầu sau khi x_p kết thúc) thì yêu cầu đó đã phải được thêm vào F theo thuật toán tham lam
- Vậy $G'' = F$, mà $|G''| = |G|$, nên F là giải pháp tối ưu

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

❑ **Cài đặt giải thuật**

```
thuexe(S)
begin
    n = length(S)
    // Sắp xếp các yêu cầu theo thời điểm kết thúc thuê tăng dần
    S = {a1, a2, ...,an} với f{a1} ≤ f{a2} ≤ ... ≤ f{an}
    F = {a1}
    i = 1
    for j from 2 to n do
        if (f(ai) ≤ s(aj)) then
            F = F ∪ {aj}
            i = j
        endif
    endfor
    return F
end
```

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 1: Thuê xe

- **Độ phức tạp của thuật toán**
  - Sắp xếp các yêu cầu
    - O(nlogn)
  - Xây dựng giải pháp
    - O(n)

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 2: Đổi tiền

- Bài toán
  - Cho một hệ thống tiền tệ gồm các loại tờ giấy tiền có mệnh giá là 1, 5, 10, 20, 50. Cần đổi một số tiền S sao cho số tờ cần dùng ít nhất.
  - Ví dụ
    - 98 = 1 + 1 + 1 + 5 + 50 + 20 + 20
  - Thuật toán đơn giản
    - liệt kê tất cả các kết hợp có thể cho tổng số tiền là S
    - chọn kết hợp dùng ít số tờ nhất
    - Độ phức tạp hàm mũ !

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 2: Đổi tiền

- Thuật toán tham lam
  - Chọn lựa tham lam
    - ở mỗi bước, chọn tờ giấy tiền có mệnh giá cao nhất có thể mà không vượt quá tổng số tiền cần đổi
  - Ví dụ: S = 98

S=98	S=48	S=28	S=8	S=3	S=3	S=3
50	20	20	5	1	1	1

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 2: Đổi tiền

- Thuật toán tham lam

```
doitien(S)
begin
   F = ∅
   if (S ≥ 50) then
      F = F ∪ {(S div 50) tờ mệnh giá 50}
      S = S mod 50
   endif
   if (S ≥ 20) then
      F = F ∪ {(S div 20) tờ mệnh giá 20}
      S = S mod 20
   endif
   if (S ≥ 10) then
      F = F ∪ {(S div 10) tờ mệnh giá 10}
      S = S mod 10
   endif
   // tương tự cho các tờ tiền mệnh giá 5, 2, 1
   ...
end
```

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 2: Đổi tiền

- Lưu ý
  - Thuật toán tham lam này không cho giải pháp tối ưu đối với mọi hệ thống tiền tệ
    - Chẳng hạn, thuật toán sẽ không cho giải pháp tối ưu đối với hệ thống tiền tệ {6, 4, 1}
    - Ví dụ
      - S = 8
      - Giải pháp cho bởi thuật toán tham lam: 6 + 1 + 1
      - Giải pháp tối ưu: 4 + 4

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 3: Xếp ba lô

- Bài toán
  - cho n đồ vật và một ba lô có trọng lượng tối đa W
  - mỗi đồ vật i có trọng lượng w_i
  - mỗi đồ vật i có giá trị v_i
  - gọi x_i là một phần của đồ vật i, $0 \le x_i \le 1$, x_i có trọng lượng $x_i w_i$ và giá trị $x_i v_i$
  - Yêu cầu: xếp các đồ vật vào ba lô để tổng giá trị ba lô lớn nhất
- Bài toán xếp ba lô này được gọi là *xếp ba lô « từng phần »*
  - có thể chỉ cần xếp vào ba lô một phần của đồ vật
- Bài toán xếp ba lô đã gặp được gọi là *xếp ba lô « 0-1 »*
  - một đồ vật hoặc được xếp vào ba lô (1) hoặc không được xếp vào ba lô (0)

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 3: Xếp ba lô

- Ý tưởng
  - Tập ứng cử viên là các đồ vật
  - Ở mỗi bước, chọn đồ vật *triển vọng nhất* và xếp vào ba lô một phần lớn nhất có thể của đồ vật này
    - đối với các đồ vật được chọn đầu tiên, xếp toàn bộ đồ vật vào ba lô
    - đối với đồ vật được chọn cuối cùng, có thể chỉ xếp một phần đồ vật vào ba lô
  - Thuật toán dừng khi ba lô đầy
  - Chọn đồ vật theo tiêu chí nào ?
    - Giá trị giảm dần
    - Trọng lượng tăng dần
    - Tỷ lệ giá trị trên trọng lượng (v_i/w_i) giảm dần
  - **Chọn lựa tham lam**: chọn đồ vật có tỷ lệ giá trị trên trọng lượng (v_i/w_i) giảm dần

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Một số ví dụ minh họa

Ví dụ 3: Xếp ba lô

- Thuật toán

```
xepbalotungphan()
begin
    sắp xếp các đồ vật theo tỷ lệ v_i/w_i giảm dần
    w = W
    i = 1
    while (w ≥ w_i) do // xếp toàn bộ đồ vật vào ba lô
        x_i = 1
        w = w – w_i
        i = i +1
    endwhile
    x_i = w_i/w // đồ vật cuối cùng được chọn để xếp vào ba lô
    for k from i + 1 to n do
        x_i = 0 // các đồ vật không được xếp vào bao lô
    endfor
    return (x_1, x_2, …,x_n) // giải pháp
end
```

- Phân tích độ phức tạp
  - Sắp xếp: O(nlogn)
  - Lặp: duyệt n đồ vật, vậy O(n)

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Nhận xét (tính chất của thuật toán Tham lam)

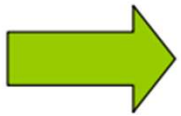
- Thuật toán tham lam
  - Xác định giải pháp sau một dãy liên tiếp các lựa chọn
  - Mỗi bước quyết định, lựa chọn dường như tốt nhất ở bước đó sẽ được chọn
  - Không luôn cho giải pháp tối ưu
- Một bài toán tối ưu bất kỳ có thể được giải quyết bởi thuật toán tham lam ?
  - Không luôn luôn đúng
- Tuy nhiên, nếu một bài toán có hai tính chất
  - Tính chất chọn lựa tham lam
  - Cấu trúc con tối ưu
- Thì bài toán có thể giải quyết được bởi thuật toán tham lam

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Nhận xét (tính chất của thuật toán Tham lam)

- Tính chất chọn lựa tham lam (greedy choice property)
  - Luôn tồn tại một giải pháp tối ưu chứa một chọn lựa tham lam
    - Cần chỉ ra tồn tại một giải pháp tối ưu luôn bắt đầu bởi một chọn lựa tham lam

- Tính chất cấu trúc con tối ưu (optimal substructure)
  - Nếu *F* là một giải pháp tối ưu chứa một chọn lựa tham lam *c* thì *F*-{*c*} là giải pháp tối ưu cho bài toán con tương tự như bài toán đầu không chứa *c*
    - Giải pháp tối ưu của một bài toán *phải* chứa giải pháp tối ưu của bài toán con của nó



Chứng minh tính tối ưu của thuật toán tham lam

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Nhận xét (tính chất của thuật toán Tham lam)

- Nếu một bài toán thoả mãn hai tính chất
  - Tính chất chọn lựa tham lam
  - Tính chất cấu trúc con tối ưu
- Thì thuật toán tham lam cho giải pháp tối ưu

- Chứng minh
  - Theo tính chất chọn lựa tham lam, tồn tại giải pháp tối ưu S chứa một chọn lựa tham lam c_1. Theo tính chất cấu trúc con tối ưu, $F-\{c_1\}$ là giải pháp tối ưu của bài toán con không chứa c_1.
  - Áp dụng cho bài toán con không chứa c_1, theo tính chất chọn lựa tham lam, $F-\{c_1\}$ là giải pháp tối ưu chứa chọn lựa tham lam c_2. Theo tính chất cấu trúc con tối ưu, $F-\{c_1, c_2\}$ là giải pháp tối ưu cho bài toán con không chứa c_1 và c_2.
  - Tiếp tục lý giải như thế, cuối cùng chúng ta có
    $$F-\{c_1, c_2, \ldots, c_n\} = \varnothing$$
  - Hay: $F = \{c_1, c_2, \ldots, c_n\}$
  - Vậy giải pháp tối ưu F của bài toán ban đầu là một dãy các lựa chọn tham lam thực hiện bởi thuật toán tham lam

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

So sánh với qui hoạch động

Đặc điểm	Greedy	DP
Ghi nhớ trạng thái	❌ Không	✅ Có
Quay lui	❌ Không	✅ Có
Hiệu quả	✅ Rất nhanh	❌ Thường chậm hơn
Điều kiện áp dụng	Chặt chẽ hơn	Linh hoạt hơn

GIẢI THUẬT THAM LAM (GREEDY ALGORITHM)

Bài tập tự học

Cài đặt các bài toán băng giải thuật Greedy Algorithm

- Đổi tiền
- Xếp ba lô "0-1"