Lập luận tráo đổi

Bùi Việt Dũng

Ngày 14 tháng 10 năm 2020

Bài toán tối ưu tổ hợp

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.
    - *Nghiệm* ở đây có nghĩa là một lời giải có thể của bài toán, không phải kết quả tối ưu của bài toán.

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.
  - *Nghiệm* ở đây có nghĩa là một lời giải có thể của bài toán, không phải kết quả tối ưu của bài toán.
  - Ví dụ về bài toán tối ưu tổ hợp: Tìm cách chọn k vật trong số n vật sao cho tổng giá trị của k vật được chọn là lớn nhất.

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.
  - *Nghiệm* ở đây có nghĩa là một lời giải có thể của bài toán, không phải kết quả tối ưu của bài toán.
  - Ví dụ về bài toán tối ưu tổ hợp: Tìm cách chọn k vật trong số n vật sao cho tổng giá trị của k vật được chọn là lớn nhất.
  - Ví dụ về các bài toán không phải bài toán tối ưu tổ hợp: Tìm giá trị lớn nhất của $f(x)$ trên $\mathbb{R}$.

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.
  - *Nghiệm* ở đây có nghĩa là một lời giải có thể của bài toán, không phải kết quả tối ưu của bài toán.
  - Ví dụ về bài toán tối ưu tổ hợp: Tìm cách chọn k vật trong số n vật sao cho tổng giá trị của k vật được chọn là lớn nhất.
  - Ví dụ về các bài toán không phải bài toán tối ưu tổ hợp: Tìm giá trị lớn nhất của $f(x)$ trên $\mathbb{R}$.
- Bài toán tối ưu tổ hợp thì luôn có nghiệm tối ưu

Bài toán tối ưu tổ hợp

- Một bài toán được gọi là bài toán tối ưu tổ hợp nếu bài toán đó số *nghiệm* hữu hạn.
  - *Nghiệm* ở đây có nghĩa là một lời giải có thể của bài toán, không phải kết quả tối ưu của bài toán.
  - Ví dụ về bài toán tối ưu tổ hợp: Tìm cách chọn k vật trong số n vật sao cho tổng giá trị của k vật được chọn là lớn nhất.
  - Ví dụ về các bài toán không phải bài toán tối ưu tổ hợp: Tìm giá trị lớn nhất của $f(x)$ trên $\mathbb{R}$.
- Bài toán tối ưu tổ hợp thì luôn có nghiệm tối ưu
  - Ta luôn có thể tìm được *nghiệm tối ưu* bằng cách sinh toàn bộ nghiệm của bài toán, rồi xét lần lượt từng nghiệm một.

Phương pháp chứng minh thuật tham

Phương pháp chứng minh thuật tham

Bước 1. Tìm một đặc điểm x nào đó để tham.

Phương pháp chứng minh thuật tham

Bước 1. Tìm một đặc điểm x nào đó để tham.

Bước 2. Chứng minh rằng nếu tồn tại nghiệm tối ưu X' không thỏa mãn đặc điểm x, ta luôn dựng được nghiệm X thỏa mãn đặc điểm tham sao cho nghiệm X không tệ hơn nghiệm tối ưu X' (tức X cũng là nghiệm tối ưu).

Phương pháp chứng minh thuật tham

Bước 1. Tìm một đặc điểm x nào đó để tham.

Bước 2. Chứng minh rằng nếu tồn tại nghiệm tối ưu X' không thỏa mãn đặc điểm x, ta luôn dựng được nghiệm X thỏa mãn đặc điểm tham sao cho nghiệm X không tệ hơn nghiệm tối ưu X' (tức X cũng là nghiệm tối ưu).

- Gọi tập S là tập các nghiệm tối ưu. Do bài toán ta đang xét là bài toán tối ưu tổ hợp, $S \neq \emptyset$. Chọn một nghiệm O bất kì thuộc S.
  - Nếu O thỏa mãn đặc điểm x, ta tìm được nghiệm tối ưu $O' = O \in S$ thỏa mãn đặc điểm x.
  - Nếu O không thỏa mãn đặc điểm x, từ chứng minh ở bước 2, ta có thể dựng một nghiệm O' tối ưu thỏa mãn đặc điểm x và nằm trong S.

  Do đó, ta luôn tìm được một nghiệm tối ưu thỏa mãn đặc điểm x nằm trong S. Nói cách khác, nếu ta chỉ xét các nghiệm thỏa mãn đặc điểm x thì ta vẫn sẽ tìm được một nghiệm tối ưu.

Hai phương pháp chứng minh

Hai phương pháp chứng minh

- Lập luận Tham lam luôn đi đầu ("Greedy Stays Ahead" Argument): Thường dùng cho các bài mà nghiệm là một chỉnh hợp của các đối tượng được cho (ví dụ: chọn 2 vật trong 3 vật).

Hai phương pháp chứng minh

- Lập luận Tham lam luôn đi đầu ("Greedy Stays Ahead" Argument): Thường dùng cho các bài mà nghiệm là một chỉnh hợp của các đối tượng được cho (ví dụ: chọn 2 vật trong 3 vật).
- $\boxed{\text{Lập luận Tráo đổi (Exchange Argument)}}$: Thường được dùng cho các bài mà nghiệm là một hoán vị của các đối tượng được cho (ví dụ: sắp xếp thứ tự n công việc sao cho thời gian thực hiện là nhỏ nhất).

Cho hai dãy số $a_1, a_2, ..., a_n$ và $b_1, b_2, ..., b_n$. Tìm cách sắp xếp lại hai dãy số này sao cho $a_1 b_1 + a_2 b_2 + ... + a_n b_n$ là nhỏ nhất.
Giới hạn: $1 \leq n \leq 800; |a_i|, |b_i| \leq 10^5$

Cho hai dãy số $a_1, a_2, ..., a_n$ và $b_1, b_2, ..., b_n$. Tìm cách sắp xếp lại hai dãy số này sao cho $a_1 b_1 + a_2 b_2 + ... + a_n b_n$ là nhỏ nhất.
Giới hạn: $1 \le n \le 800; |a_i|, |b_i| \le 10^5$
Ví dụ:
Nếu

$$a = (1, 3, -5) \text{ và } b = (-2, 4, 1)$$

thì ta có thể sắp xếp hai dãy thành

$$a = (-5, 1, 3) \text{ và } b = (4, 1, -2)$$

để nhận được kết quả

$$-5 \times 4 + 1 \times 1 + 3 \times (-2) = -25$$

nhỏ nhất

Hướng làm

Hướng làm

Đầu tiên, nhận thấy mục tiêu là tìm tổng nhỏ nhất, ta nghĩ ngay đến việc các số âm phải được ghép với số dương $\rightarrow$ Lời giải 1

Hướng làm

Đầu tiên, nhận thấy mục tiêu là tìm tổng nhỏ nhất, ta nghĩ ngay đến việc các số âm phải được ghép với số dương $\rightarrow$ Lời giải 1

- Sắp xếp dãy a theo thứ tự tăng dần, sắp xếp dãy b theo thứ tự giảm dần.
- Tổng $a_1b_1 + a_2b_2 + ... + a_nb_n$ lúc này sẽ nhỏ nhất.
- Độ phức tạp $O(n \log n)$

Hướng làm

Đầu tiên, nhận thấy mục tiêu là tìm tổng nhỏ nhất, ta nghĩ ngay đến việc các số âm phải được ghép với số dương $\rightarrow$ Lời giải 1

- Sắp xếp dãy a theo thứ tự tăng dần, sắp xếp dãy b theo thứ tự giảm dần.
- Tổng $a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$ lúc này sẽ nhỏ nhất.
- Độ phức tạp $O(n \log n)$

Vấn đề: Tại sao giới hạn bài này lại là $n \leq 800 \rightarrow$ Lời giải 2

Hướng làm

Đầu tiên, nhận thấy mục tiêu là tìm tổng nhỏ nhất, ta nghĩ ngay đến việc các số âm phải được ghép với số dương $\rightarrow$ Lời giải 1

- Sắp xếp dãy a theo thứ tự tăng dần, sắp xếp dãy b theo thứ tự giảm dần.
- Tổng $a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$ lúc này sẽ nhỏ nhất.
- Độ phức tạp $O(n \log n)$

Vấn đề: Tại sao giới hạn bài này lại là $n \leq 800$ $\rightarrow$ Lời giải 2

- Tạo đồ thị hai phía, mỗi phía một dãy.
- Với mỗi cặp hai số a_i, b_j, tạo một cạnh có trọng số $a_i b_j$ nối từ a_i đến b_j.
- Dùng luồng cực đại với chi phí cực tiểu để tìm đáp án.
- Độ phức tạp $O(n^3)$ với hằng số thấp, khá phù hợp với giới hạn $n \leq 800$.

Tình huống

Tình huống

- Trong các kì thi chỉ được phép nộp một lần (thi HSG QG, COCI, Google Code Jam, ...), nếu còn thừa thời gian thì sau khi làm lời giải 1 thì liệu ta có nên viết thêm thuật trâu hay làm thêm lời giải 2 cho chắc?

Tình huống

- Trong các kì thi chỉ được phép nộp một lần (thi HSG QG, COCI, Google Code Jam, ...), nếu còn thừa thời gian thì sau khi làm lời giải 1 thì liệu ta có nên viết thêm thuật trâu hay làm thêm lời giải 2 cho chắc?
- Giả sử khi đang thi ICPC, bạn sẽ làm thế nào để thuyết phục đồng đội đưa máy cho mình?

Tình huống

- Trong các kì thi chỉ được phép nộp một lần (thi HSG QG, COCI, Google Code Jam, ...), nếu còn thừa thời gian thì sau khi làm lời giải 1 thì liệu ta có nên viết thêm thuật trâu hay làm thêm lời giải 2 cho chắc?
- Giả sử khi đang thi ICPC, bạn sẽ làm thế nào để thuyết phục đồng đội đưa máy cho mình?
  Và nếu nhận được máy, bạn lập trình Lời giải 1 và nhận về kết quả "Đáp án sai (WA)" thì đó là do bạn lập trình lỗi hay là do lời giải 1 sai thật?

Tình huống

- Trong các kì thi chỉ được phép nộp một lần (thi HSG QG, COCI, Google Code Jam, ...), nếu còn thừa thời gian thì sau khi làm lời giải 1 thì liệu ta có nên viết thêm thuật trâu hay làm thêm lời giải 2 cho chắc?
- Giả sử khi đang thi ICPC, bạn sẽ làm thế nào để thuyết phục đồng đội đưa máy cho mình?
  Và nếu nhận được máy, bạn lập trình Lời giải 1 và nhận về kết quả "Đáp án sai (WA)" thì đó là do bạn lập trình lỗi hay là do lời giải 1 sai thật?

$\Rightarrow$ Việc chứng minh lời giải 1 rất quan trọng.

Lập luận Tráo đổi

Lập luận Tráo đổi

Để chứng minh một thuật tham lam được cho trước, ta làm theo 4 bước sau.

Lập luận Tráo đổi

Để chứng minh một thuật tham lam được cho trước, ta làm theo 4 bước sau.

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Lập luận Tráo đổi

Để chứng minh một thuật tham lam được cho trước, ta làm theo 4 bước sau.

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 2. Từ thuật toán được cho, xác định đặc điểm của một nghiệm tối ưu O của bài toán.

Để chứng minh một thuật tham lam được cho trước, ta làm theo 4 bước sau.

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 2. Từ thuật toán được cho, xác định đặc điểm của một nghiệm tối ưu O của bài toán.

Bước 3. Dùng phản chứng để chứng minh nếu nghiệm tối ưu F không thỏa mãn đặc điểm đã xác định ở bước 2 thì tồn tại nghiệm F' gần giống nghiệm O hơn F.

Để chứng minh một thuật tham lam được cho trước, ta làm theo 4 bước sau.

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 2. Từ thuật toán được cho, xác định đặc điểm của một nghiệm tối ưu O của bài toán.

Bước 3. Dùng phản chứng để chứng minh nếu nghiệm tối ưu F không thỏa mãn đặc điểm đã xác định ở bước 2 thì tồn tại nghiệm F' gần giống nghiệm O hơn F.

Bước 4. Chứng minh rằng có cách biến đổi F thành O sau hữu hạn bước tráo đổi.

Bước 1: Xác định X, $d(X)$

- Ta thấy thứ tự dãy a không quan trọng nên để đơn giản hóa bài toán, ta coi a đã được sắp xếp theo thứ tự tăng dần.

Bước 1: Xác định X, $d(X)$

- Ta thấy thứ tự dãy a không quan trọng nên để đơn giản hóa bài toán, ta coi a đã được sắp xếp theo thứ tự tăng dần.
- Ta có thể biểu diễn một nghiệm X của bài toán bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$ với ý nghĩa số a_i sẽ được "ghép" với số b_{x_i}.

Bước 1: Xác định X, $d(X)$

- Ta thấy thứ tự dãy a không quan trọng nên để đơn giản hóa bài toán, ta coi a đã được sắp xếp theo thứ tự tăng dần.
- Ta có thể biểu diễn một nghiệm X của bài toán bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$ với ý nghĩa số a_i sẽ được "ghép" với số b_{x_i}.
- $d(X) = \sum_{i=1}^{n} a_i b_{x_i}$
  $d(X)$ càng nhỏ thì X càng tốt.

Bước 2: Xác định đặc điểm O

Bước 2: Xác định đặc điểm O

- Điều phải chứng minh: Nếu $O = o_1, o_2, ..., o_n$ thỏa mãn $b_{o_1} \geq b_{o_2} \geq ... \geq b_n$ thì O là một nghiệm tối ưu.

Bước 3: Giả sử tồn tại F tối ưu

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.
- Khi đó, tồn tại i sao cho $b_{f_i} < b_{f_{i+1}}$ (vì sao?). Chọn i sao cho i nhỏ nhất có thể .

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.
- Khi đó, tồn tại i sao cho $b_{f_i} < b_{f_{i+1}}$ (vì sao?). Chọn i sao cho i nhỏ nhất có thể .
- Tráo đổi f_i và f_{i+1} để nhận được nghiệm $F' = f_1, f_2, ..., f_{i-1}, f_{i+1}, f_i, f_{i+2}, ..., f_n$. Ta cần chứng minh rằng F' sẽ không kém hơn F, hay $d(F') \leq d(F)$

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.
- Khi đó, tồn tại i sao cho $b_{f_i} < b_{f_{i+1}}$ (vì sao?). Chọn i sao cho i nhỏ nhất có thể .
- Tráo đổi f_i và f_{i+1} để nhận được nghiệm $F' = f_1, f_2, ..., f_{i-1}, f_{i+1}, f_i, f_{i+2}, ..., f_n$. Ta cần chứng minh rằng F' sẽ không kém hơn F, hay $d(F') \leq d(F)$
- $d(F') = \sum_{i=1}^{i-1} a_i b_{f_i} + a_i b_{f_{i+1}} + a_{i+1} b_{f_i} + \sum_{i=i+2}^{n} a_i b_{f_i}$

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.
- Khi đó, tồn tại i sao cho $b_{f_i} < b_{f_{i+1}}$ (vì sao?). Chọn i sao cho i nhỏ nhất có thể .
- Tráo đổi f_i và f_{i+1} để nhận được nghiệm $F' = f_1, f_2, ..., f_{i-1}, f_{i+1}, f_i, f_{i+2}, ..., f_n$. Ta cần chứng minh rằng F' sẽ không kém hơn F, hay $d(F') \leq d(F)$
- $d(F') = \sum_{i=1}^{i-1} a_i b_{f_i} + a_i b_{f_{i+1}} + a_{i+1} b_{f_i} + \sum_{i=i+2}^{n} a_i b_{f_i}$
- $d(F) = \sum_{i=1}^{i-1} a_i b_{f_i} + a_i b_{f_i} + a_{i+1} b_{f_{i+1}} + \sum_{i=i+2}^{n} a_i b_{f_i}$

Bước 3: Giả sử tồn tại F tối ưu

- Giả sử tồn tại nghiệm tối ưu $F = f_1, f_2, ..., f_n$ không thỏa mãn $b_{f_1} \geq b_{f_2} \geq ... \geq b_{f_n}$.
- Khi đó, tồn tại i sao cho $b_{f_i} < b_{f_{i+1}}$ (vì sao?). Chọn i sao cho i nhỏ nhất có thể .
- Tráo đổi f_i và f_{i+1} để nhận được nghiệm $F' = f_1, f_2, ..., f_{i-1}, f_{i+1}, f_i, f_{i+2}, ..., f_n$. Ta cần chứng minh rằng F' sẽ không kém hơn F, hay $d(F') \leq d(F)$
- $d(F') = \sum_{i=1}^{i-1} a_i b_{f_i} + a_i b_{f_{i+1}} + a_{i+1} b_{f_i} + \sum_{i=i+2}^{n} a_i b_{f_i}$
- $d(F) = \sum_{i=1}^{i-1} a_i b_{f_i} + a_i b_{f_i} + a_{i+1} b_{f_{i+1}} + \sum_{i=i+2}^{n} a_i b_{f_i}$
- Xét hiệu $d(F') - d(F)$
  $d(F') - d(F) = (a_i b_{f_{i+1}} + a_{i+1} b_{f_i}) - (a_i b_{f_i} + a_{i+1} b_{f_{i+1}})$
  $d(F') - d(F) = a_i(b_{f_{i+1}} - b_{f_i}) + a_{i+1}(b_{f_i} - b_{f_{i+1}})$
  $d(F') - d(F) = (a_i - a_{i+1})(b_{f_{i+1}} - b_{f_i}) \leq 0$

Bước 4: Chứng minh có thể biến đổi F thành O

Bước 4: Chứng minh có thể biến đổi F thành O

Sử dụng sắp xếp nổi bọt với điều kiện $i < j \Rightarrow b_{f_i} \geq b_{f_j}$, ta có thể biến đổi nghiệm tối ưu F thành nghiệm O cũng tối ưu thỏa mãn đặc điểm được cho trong bước 2.

Bước 4: Chứng minh có thể biến đổi F thành O

Sử dụng sắp xếp nổi bọt với điều kiện $i < j \Rightarrow b_{f_i} \geq b_{f_j}$, ta có thể biến đổi nghiệm tối ưu F thành nghiệm O cũng tối ưu thỏa mãn đặc điểm được cho trong bước 2.

Liệu có khi nào ta không sử dụng được sắp xếp nổi bọt không?

Bước 4: Chứng minh có thể biến đổi F thành O

Sử dụng sắp xếp nổi bọt với điều kiện $i < j \Rightarrow b_{f_i} \geq b_{f_j}$, ta có thể biến đổi nghiệm tối ưu F thành nghiệm O cũng tối ưu thỏa mãn đặc điểm được cho trong bước 2.

Liệu có khi nào ta không sử dụng được sắp xếp nổi bọt không?

- Tồn tại hai phần tử a và b mà ta không thể so sánh được hai phần tử này với nhau.

Bước 4: Chứng minh có thể biến đổi F thành O

Sử dụng sắp xếp nổi bọt với điều kiện $i < j \Rightarrow b_{f_i} \geq b_{f_j}$, ta có thể biến đổi nghiệm tối ưu F thành nghiệm O cũng tối ưu thỏa mãn đặc điểm được cho trong bước 2.

Liệu có khi nào ta không sử dụng được sắp xếp nổi bọt không?

- Tồn tại hai phần tử a và b mà ta không thể so sánh được hai phần tử này với nhau.
- Tồn tại hai phần tử a, b khác nhau và hàm so sánh $<$ sao cho $a < b$ và $b < a$

Bước 4: Chứng minh có thể biến đổi F thành O

Sử dụng sắp xếp nổi bọt với điều kiện $i < j \Rightarrow b_{f_i} \geq b_{f_j}$, ta có thể biến đổi nghiệm tối ưu F thành nghiệm O cũng tối ưu thỏa mãn đặc điểm được cho trong bước 2.

Liệu có khi nào ta không sử dụng được sắp xếp nổi bọt không?

- Tồn tại hai phần tử a và b mà ta không thể so sánh được hai phần tử này với nhau.
- Tồn tại hai phần tử a, b khác nhau và hàm so sánh $<$ sao cho $a < b$ và $b < a$
- Tồn tại hai phần tử a, b, c đôi một khác nhau và hàm so sánh $<$ sao cho $a < b$, $b < c$ và $c < a$.

Cho p bài toán cổ điển có độ khó $x_1, x_2, ..., x_p$ và q bài toán sáng tạo có độ khó $y_1, y_2, ..., y_q$. Thiết kế chương trình học gồm n ($n \leq \min(p, q)$) ngày sao cho:

Cho p bài toán cổ điển có độ khó $x_1, x_2, ..., x_p$ và q bài toán sáng tạo có độ khó $y_1, y_2, ..., y_q$. Thiết kế chương trình học gồm n ($n \leq \min(p, q)$) ngày sao cho:

- Mỗi ngày học gồm đúng 1 bài cổ điển và 1 bài sáng tạo. Tổng độ khó hai bài không vượt quá s.

Cho p bài toán cổ điển có độ khó $x_1, x_2, ..., x_p$ và q bài toán sáng tạo có độ khó $y_1, y_2, ..., y_q$. Thiết kế chương trình học gồm n ($n \leq \min(p, q)$) ngày sao cho:

- Mỗi ngày học gồm đúng 1 bài cổ điển và 1 bài sáng tạo. Tổng độ khó hai bài không vượt quá s.
- Độ chênh lệnh độ khó giữa hai bài của một ngày bất kì D là nhỏ nhất có thể.

Cho p bài toán cổ điển có độ khó $x_1, x_2, ..., x_p$ và q bài toán sáng tạo có độ khó $y_1, y_2, ..., y_q$. Thiết kế chương trình học gồm n ($n \leq \min(p, q)$) ngày sao cho:

- Mỗi ngày học gồm đúng 1 bài cổ điển và 1 bài sáng tạo. Tổng độ khó hai bài không vượt quá s.
- Độ chênh lệnh độ khó giữa hai bài của một ngày bất kì D là nhỏ nhất có thể.

Giới hạn: $n, p, q \leq 2 \times 10^5; x_i, y_i, s \geq 0$

Nhận xét ban đầu

Nhận xét ban đầu

- Giới hạn n lớn $\Rightarrow$ có thể bỏ qua các hướng làm như Bộ ghép cực đại hay Quy hoạch động.

Nhận xét ban đầu

- Giới hạn n lớn $\Rightarrow$ có thể bỏ qua các hướng làm như Bộ ghép cực đại hay Quy hoạch động.
- Thứ tự sắp xếp ban đầu không làm ảnh hưởng kết quả bài toán $\Rightarrow$ sắp xếp x, y theo thứ tự tăng dần cho dễ xử lí.

Nhận xét ban đầu

- Giới hạn n lớn $\Rightarrow$ có thể bỏ qua các hướng làm như Bộ ghép cực đại hay Quy hoạch động.
- Thứ tự sắp xếp ban đầu không làm ảnh hưởng kết quả bài toán $\Rightarrow$ sắp xếp x, y theo thứ tự tăng dần cho dễ xử lí.
- D có thể tìm kiếm nhị phân được $\Rightarrow$ Ta có thể tìm cách giải bài toán với D cố định rồi tìm kiếm nhị phân D sau.

Xử lí điều kiện về s và D

Xử lí điều kiện về s và D

- Giả sử ta "ghép" bài x_i với bài y_j để dùng cho một ngày. Khi đó

Xử lí điều kiện về s và D

- Giả sử ta "ghép" bài x_i với bài y_j để dùng cho một ngày. Khi đó
  - $x_i + y_j \leq s \Rightarrow 0 \leq y_j \leq s - x_j$

Xử lí điều kiện về s và D

- Giả sử ta "ghép" bài x_i với bài y_j để dùng cho một ngày. Khi đó
  - $x_i + y_j \leq s \Rightarrow 0 \leq y_j \leq s - x_j$
  - Chênh lệch độ khó không vượt quá $D \Rightarrow x_i - D \leq y_j \leq x_i + D$

Xử lí điều kiện về s và D

- Giả sử ta "ghép" bài x_i với bài y_j để dùng cho một ngày. Khi đó
  - $x_i + y_j \leq s \Rightarrow 0 \leq y_j \leq s - x_j$
  - Chênh lệch độ khó không vượt quá $D \Rightarrow x_i - D \leq y_j \leq x_i + D$
- Như vậy nếu ta dùng bài x_i thì ta sẽ chỉ dùng được các bài y_j sao cho $\max(0, x_i - D) \leq y_j \leq \min(s - x_j, x_i + D)$

Xử lí điều kiện về s và D

- Giả sử ta "ghép" bài x_i với bài y_j để dùng cho một ngày. Khi đó
  - $x_i + y_j \leq s \Rightarrow 0 \leq y_j \leq s - x_j$
  - Chênh lệch độ khó không vượt quá $D \Rightarrow x_i - D \leq y_j \leq x_i + D$
- Như vậy nếu ta dùng bài x_i thì ta sẽ chỉ dùng được các bài y_j sao cho $\max(0, x_i - D) \leq y_j \leq \min(s - x_j, x_i + D)$
- Quy về một dạng bài Xếp lịch công việc có thể Tham lam được.

Một bài toán xếp lịch công việc

Một bài toán xếp lịch công việc

Cho n giây và n công việc, mỗi công việc chỉ có thể bắt đầu làm trong đoạn thời gian $[l_i, r_i]$, mỗi công việc làm tốn mất một giây. Tìm cách chọn các công việc để ta có thể làm được nhiều việc nhất.

Một bài toán xếp lịch công việc

Cho n giây và n công việc, mỗi công việc chỉ có thể bắt đầu làm trong đoạn thời gian $[l_i, r_i]$, mỗi công việc làm tốn mất một giây. Tìm cách chọn các công việc để ta có thể làm được nhiều việc nhất.

- Đáp án: Công việc nào hết hạn trước thì ta sẽ ưu tiên làm trước.

Một bài toán xếp lịch công việc

Cho n giây và n công việc, mỗi công việc chỉ có thể bắt đầu làm trong đoạn thời gian $[l_i, r_i]$, mỗi công việc làm tốn mất một giây. Tìm cách chọn các công việc để ta có thể làm được nhiều việc nhất.

- Đáp án: Công việc nào hết hạn trước thì ta sẽ ưu tiên làm trước.
- Ta có thể chứng minh điều này bằng Lập luận Tráo đổi.

Bước 1: Xác định $X, d(X)$

Bước 1: Xác định $X, d(X)$

- Để đơn giản hoá bài toán, sắp xếp sẵn các công việc theo thứ tự r_i tăng dần, tức $i < j \Rightarrow r_i < r_j$.

Bước 1: Xác định $X, d(X)$

- Để đơn giản hoá bài toán, sắp xếp sẵn các công việc theo thứ tự r_i tăng dần, tức $i < j \Rightarrow r_i < r_j$.
- Ta có thể biểu diễn nghiệm X bằng một dãy n sự kiện $x_1, x_2, ..., x_n$. Nếu $x_i = -1$ thì ta không làm gì ở giây thứ i, còn nếu $1 \le x_i \le n$ thì ta làm công việc $[l_{x_i}, r_{x_i}]$ ở giây thứ i. Ngoài ra,

Bước 1: Xác định $X, d(X)$

- Để đơn giản hoá bài toán, sắp xếp sẵn các công việc theo thứ tự r_i tăng dần, tức $i < j \Rightarrow r_i < r_j$.
- Ta có thể biểu diễn nghiệm X bằng một dãy n sự kiện $x_1, x_2, ..., x_n$. Nếu $x_i = -1$ thì ta không làm gì ở giây thứ i, còn nếu $1 \leq x_i \leq n$ thì ta làm công việc $[l_{x_i}, r_{x_i}]$ ở giây thứ i. Ngoài ra,
  - $\forall 1 \leq i \leq n : l_{x_i} \leq i \leq r_{x_i}$ (công việc phải làm trong hạn).

Bước 1: Xác định $X, d(X)$

- Để đơn giản hoá bài toán, sắp xếp sẵn các công việc theo thứ tự r_i tăng dần, tức $i < j \Rightarrow r_i < r_j$.
- Ta có thể biểu diễn nghiệm X bằng một dãy n sự kiện $x_1, x_2, ..., x_n$. Nếu $x_i = -1$ thì ta không làm gì ở giây thứ i, còn nếu $1 \leq x_i \leq n$ thì ta làm công việc $[l_{x_i}, r_{x_i}]$ ở giây thứ i. Ngoài ra,
  - $\forall 1 \leq i \leq n : l_{x_i} \leq i \leq r_{x_i}$ (công việc phải làm trong hạn).
  - $i \neq j \Rightarrow x_i \neq x_j$ (không công việc nào được làm 2 lần)

Bước 1: Xác định $X, d(X)$

- Để đơn giản hoá bài toán, sắp xếp sẵn các công việc theo thứ tự r_i tăng dần, tức $i < j \Rightarrow r_i < r_j$.
- Ta có thể biểu diễn nghiệm X bằng một dãy n sự kiện $x_1, x_2, ..., x_n$. Nếu $x_i = -1$ thì ta không làm gì ở giây thứ i, còn nếu $1 \leq x_i \leq n$ thì ta làm công việc $[l_{x_i}, r_{x_i}]$ ở giây thứ i. Ngoài ra,
  - $\forall 1 \leq i \leq n : l_{x_i} \leq i \leq r_{x_i}$ (công việc phải làm trong hạn).
  - $i \neq j \Rightarrow x_i \neq x_j$ (không công việc nào được làm 2 lần)
- $d(X) = \sum_{i=1}^{n} \mathbb{1}[x_i \neq -1]$

Bước 2: Xác định đặc điểm O

Bước 2: Xác định đặc điểm O

- Ta cần chứng minh tồn tại nghiệm $O = o_1, o_2, ..., o_n$ thỏa mãn $\forall 1 \leq i \leq n$:

Bước 2: Xác định đặc điểm O

- Ta cần chứng minh tồn tại nghiệm $O = o_1, o_2, ..., o_n$ thỏa mãn $\forall 1 \leq i \leq n$:
  - $l_{o_i} \leq i \leq r_{o_i}$

Bước 2: Xác định đặc điểm O

- Ta cần chứng minh tồn tại nghiệm $O = o_1, o_2, ..., o_n$ thỏa mãn $\forall 1 \leq i \leq n$:
  - $l_{o_i} \leq i \leq r_{o_i}$
  - Trong số các công việc thỏa mãn $l_j \leq i \leq r_j$ và chưa được chọn ở $o_1, o_2, ..., o_{i-1}$, r_{o_i} có giá trị nhỏ nhất.

Bước 3: Giả sử tồn tại F tối ưu

Bước 3: Giả sử tồn tại F tối ưu

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn đặc điểm O. Khi đó tồn tại i sao cho $r_{f_i} > r_{o_i}$. Chọn i sao cho i nhỏ nhất có thể. Ta thấy:

Bước 3: Giả sử tồn tại F tối ưu

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn đặc điểm O. Khi đó tồn tại i sao cho $r_{f_i} > r_{o_i}$. Chọn i sao cho i nhỏ nhất có thể. Ta thấy:

- Các công việc trước i trong F và trong O là giống nhau.

Bước 3: Giả sử tồn tại F tối ưu

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn đặc điểm O. Khi đó tồn tại i sao cho $r_{f_i} > r_{o_i}$. Chọn i sao cho i nhỏ nhất có thể. Ta thấy:

- Các công việc trước i trong F và trong O là giống nhau.
- Ta có thể thay f_i bằng o_i để nhận được nghiệm F' có $d(F') = d(F) \Rightarrow d(F') \leq d(F)$

Bước 4: Chứng minh có thể biến đổi F thành O

Bước 4: Chứng minh có thể biến đổi F thành O

Ta thấy ta có thể thay dần dần các công việc của F bằng công việc tương ứng của O để nhận được O. Do đó, ta hoàn toàn có thể biến đổi F thành O trong hữu hạn bước.

Fight Against Monsters

Người anh hùng được cho n con quỷ, mỗi con quỷ có 2 chỉ số tấn công a_i và máu x_i. Mỗi giây tranh đấu, hai sự kiện sau lần lượt diễn ra:

Người anh hùng được cho n con quỷ, mỗi con quỷ có 2 chỉ số tấn công a_i và máu x_i. Mỗi giây tranh đấu, hai sự kiện sau lần lượt diễn ra:

- Tất cả các con quỷ còn sống tấn công người anh hùng. Người anh hùng bị mất (tổng a_i của các con quỷ còn sống) máu.

Người anh hùng được cho n con quỷ, mỗi con quỷ có 2 chỉ số tấn công a_i và máu x_i. Mỗi giây tranh đấu, hai sự kiện sau lần lượt diễn ra:

- Tất cả các con quỷ còn sống tấn công người anh hùng. Người anh hùng bị mất (tổng a_i của các con quỷ còn sống) máu.
- Người anh hùng đánh một con quỷ. Con quỷ đó mất một máu. Nếu sau khi bị đánh, x của con quỷ đó bằng 0 thì con quỷ đó chết.

Fight Against Monsters

Người anh hùng được cho n con quỷ, mỗi con quỷ có 2 chỉ số tấn công a_i và máu x_i. Mỗi giây tranh đấu, hai sự kiện sau lần lượt diễn ra:

- Tất cả các con quỷ còn sống tấn công người anh hùng. Người anh hùng bị mất (tổng a_i của các con quỷ còn sống) máu.
- Người anh hùng đánh một con quỷ. Con quỷ đó mất một máu. Nếu sau khi bị đánh, x của con quỷ đó bằng 0 thì con quỷ đó chết.

Tìm lượng máu ít nhất mà người anh hùng cần phải mất để tiêu diệt n con quỷ.

Fight Against Monsters

Người anh hùng được cho n con quỷ, mỗi con quỷ có 2 chỉ số tấn công a_i và máu x_i. Mỗi giây tranh đấu, hai sự kiện sau lần lượt diễn ra:

- Tất cả các con quỷ còn sống tấn công người anh hùng. Người anh hùng bị mất (tổng a_i của các con quỷ còn sống) máu.
- Người anh hùng đánh một con quỷ. Con quỷ đó mất một máu. Nếu sau khi bị đánh, x của con quỷ đó bằng 0 thì con quỷ đó chết.

Tìm lượng máu ít nhất mà người anh hùng cần phải mất để tiêu diệt n con quỷ.
Giới hạn: $1 \leq n \leq 10^5; a_i, h_i \geq 1$

Nhận xét cơ bản

Nhận xét cơ bản

- Một khi ta chọn đánh con quỷ nào thì ta phải đánh con quỷ đó liên tục cho đến chết (dễ dàng chứng minh bằng lập luận tráo đổi)

Nhận xét cơ bản

- Một khi ta chọn đánh con quỷ nào thì ta phải đánh con quỷ đó liên tục cho đến chết (dễ dàng chứng minh bằng lập luận tráo đổi)
  - Trường hợp có 2 con quái (trường hợp nhỏ): Giả sử đang đánh con quái thứ nhất thì mình đánh con quái thứ hai. Con quái thứ hai không chết muộn hơn, còn con quái thứ nhất thì chắc chắn chết muộn hơn $\Rightarrow$ người anh hùng mất nhiều máu hơn.

Nhận xét cơ bản

- Một khi ta chọn đánh con quỷ nào thì ta phải đánh con quỷ đó liên tục cho đến chết (dễ dàng chứng minh bằng lập luận tráo đổi)
    - Trường hợp có 2 con quái (trường hợp nhỏ): Giả sử đang đánh con quái thứ nhất thì mình đánh con quái thứ hai. Con quái thứ hai không chết muộn hơn, còn con quái thứ nhất thì chắc chắn chết muộn hơn $\Rightarrow$ người anh hùng mất nhiều máu hơn.
- Thứ tự sắp xếp các con quỷ phải phụ thuộc vào cả a lẫn x

Nhận xét cơ bản

- Một khi ta chọn đánh con quỷ nào thì ta phải đánh con quỷ đó liên tục cho đến chết (dễ dàng chứng minh bằng lập luận tráo đổi)
    - Trường hợp có 2 con quái (trường hợp nhỏ): Giả sử đang đánh con quái thứ nhất thì mình đánh con quái thứ hai. Con quái thứ hai không chết muộn hơn, còn con quái thứ nhất thì chắc chắn chết muộn hơn $\Rightarrow$ người anh hùng mất nhiều máu hơn.
- Thứ tự sắp xếp các con quỷ phải phụ thuộc vào cả a lẫn x
    - Nếu chỉ dùng a để sắp xếp các con quái, ta có thể cho hết tất cả các a bằng nhau, rồi lấy x ngẫu nhiên. Dễ thấy trong nhiều trường hợp, thứ tự sắp xếp sẽ không tối ưu. Tương tự với trường hợp chỉ dùng x để sắp xếp các con quái.

Lập luận Đổi tráo

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Lập luận Đổi tráo

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 3* Tìm điều kiện để nếu nghiệm tối ưu F không thỏa mãn đặc điểm gì đó mà mình chưa biết thì tồn tại F' thỏa mãn đặc điểm đó hơn. Cụ thể là ta xét hiệu $d(F') - d(F)$

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 3* Tìm điều kiện để nếu nghiệm tối ưu F không thỏa mãn đặc điểm gì đó mà mình chưa biết thì tồn tại F' thỏa mãn đặc điểm đó hơn. Cụ thể là ta xét hiệu $d(F') - d(F)$

Bước 2* Xác định đặc điểm của một nghiệm tối ưu O của bài toán.

Lập luận Đổi tráo

Bước 1. Xác định cách biểu diễn một nghiệm X của bài toán, hàm tính độ tốt của một nghiệm $d(X)$

Bước 3* Tìm điều kiện để nếu nghiệm tối ưu F không thỏa mãn đặc điểm gì đó mà mình chưa biết thì tồn tại F' thỏa mãn đặc điểm đó hơn. Cụ thể là ta xét hiệu $d(F') - d(F)$

Bước 2* Xác định đặc điểm của một nghiệm tối ưu O của bài toán.

Bước 4. Dùng lập luận tráo đổi chứng minh đặc điểm tìm được luôn cho ra nghiệm tối ưu.

Bước 1: Xác định X, $d(X)$

Bước 1: Xác định X, $d(X)$

- Ta có thể biểu diễn thứ tự đánh quái X bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$

Bước 1: Xác định X, $d(X)$

- Ta có thể biểu diễn thứ tự đánh quái X bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$
- Mỗi con quái i làm người anh hùng mất a_i(Thời gian nó còn sống) $= a_i \sum_{j=1}^{i} x_i$ đơn vị máu

Bước 1: Xác định X, $d(X)$

- Ta có thể biểu diễn thứ tự đánh quái X bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$
- Mỗi con quái i làm người anh hùng mất a_i(Thời gian nó còn sống) $= a_i \sum_{j=1}^{i} x_i$ đơn vị máu
- Vậy $d(X)$
$= \sum_{i=1}^{n}$(Lượng máu người anh hùng bị mất bởi con quái i)
$= \sum_{i=1}^{n} \left(a_i \sum_{j=1}^{i} x_j \right)$

Bước 1: Xác định X, $d(X)$

- Ta có thể biểu diễn thứ tự đánh quái X bằng một hoán vị $x_1, x_2, ..., x_n$ của $1, 2, ..., n$
- Mỗi con quái i làm người anh hùng mất a_i(Thời gian nó còn sống) $= a_i \sum_{j=1}^{i} x_i$ đơn vị máu
- Vậy $d(X)$ $= \sum_{i=1}^{n}$(Lượng máu người anh hùng bị mất bởi con quái i) $= \sum_{i=1}^{n} \left(a_i \sum_{j=1}^{i} x_j \right)$
- $d(X)$ càng nhỏ thì càng tốt.

Bước 2: Xét hiệu $d(F') - d(F)$

Bước 2: Xét hiệu $d(F') - d(F)$

Giả sử F không thỏa mãn điều kiện gì đó của F'. Giả sử thêm điều kiện này liên quan đến 2 phần tử z và $z+1$ liên tiếp nằm cạnh nhau, và ta chỉ cần tráo hai phần tử này là $F = F'$.

Bước 2: Xét hiệu $d(F') - d(F)$

Giả sử F không thỏa mãn điều kiện gì đó của F'. Giả sử thêm điều kiện này liên quan đến 2 phần tử z và $z+1$ liên tiếp nằm cạnh nhau, và ta chỉ cần tráo hai phần tử này là $F = F'$.
Nói cách khác, đánh số lại các con quỷ theo F: nếu
$F = 1, 2, ..., z, z+1, ..., n$ thì $F' = 1, 2, ..., z+1, z, ..., n$. Khi đó:

Bước 2: Xét hiệu $d(F') - d(F)$

Giả sử F không thỏa mãn điều kiện gì đó của F'. Giả sử thêm điều kiện này liên quan đến 2 phần tử z và $z+1$ liên tiếp nằm cạnh nhau, và ta chỉ cần tráo hai phần tử này là $F = F'$.

Nói cách khác, đánh số lại các con quỷ theo F: nếu $F = 1, 2, ..., z, z+1, ..., n$ thì $F' = 1, 2, ..., z+1, z, ..., n$. Khi đó:

- $d(F) = \sum_{i=1}^{z-1}\left(a_i \sum_{j=1}^{i} x_j\right) + a_z\left(\sum_{j=1}^{z-1} x_j + x_z\right) + a_{z+1}(\sum_{j=1}^{z-1} x_j + x_z + x_{z+1}\Big) + \sum_{i=z+2}^{n}\left(a_i \sum_{j=1}^{i} x_j\right)$

Bước 2: Xét hiệu $d(F') - d(F)$

Giả sử F không thỏa mãn điều kiện gì đó của F'. Giả sử thêm điều kiện này liên quan đến 2 phần tử z và $z+1$ liên tiếp nằm cạnh nhau, và ta chỉ cần tráo hai phần tử này là $F = F'$.
Nói cách khác, đánh số lại các con quỷ theo F: nếu $F = 1, 2, ..., z, z+1, ..., n$ thì $F' = 1, 2, ..., z+1, z, ..., n$. Khi đó:

- $d(F) = \sum_{i=1}^{z-1}\left(a_i \sum_{j=1}^{i} x_j\right) + a_z\left(\sum_{j=1}^{z-1} x_j + x_z\right) + a_{z+1}(\sum_{j=1}^{z-1} x_j + x_z + x_{z+1}\Big) + \sum_{i=z+2}^{n}\left(a_i \sum_{j=1}^{i} x_j\right)$
- $d(F') = \sum_{i=1}^{z-1}\left(a_i \sum_{j=1}^{i} x_j\right) + a_{z+1}\left(\sum_{j=1}^{z-1} x_j + x_{z+1}\right) + a_z(\sum_{j=1}^{z-1} x_j + x_{z+1} + x_z\Big) + \sum_{i=z+2}^{n}\left(a_i \sum_{j=1}^{i} x_j\right)$

Bước 2: Xét hiệu $d(F') - d(F)$

Giả sử F không thỏa mãn điều kiện gì đó của F'. Giả sử thêm điều kiện này liên quan đến 2 phần tử z và $z+1$ liên tiếp nằm cạnh nhau, và ta chỉ cần tráo hai phần tử này là $F = F'$.

Nói cách khác, đánh số lại các con quỷ theo F: nếu $F = 1, 2, ..., z, z+1, ..., n$ thì $F' = 1, 2, ..., z+1, z, ..., n$. Khi đó:

- $d(F) = \sum_{i=1}^{z-1} \left(a_i \sum_{j=1}^{i} x_j \right) + a_z \left(\sum_{j=1}^{z-1} x_j + x_z \right) + a_{z+1} (\sum_{j=1}^{z-1} x_j + x_z + x_{z+1} \Big) + \sum_{i=z+2}^{n} \left(a_i \sum_{j=1}^{i} x_j \right)$
- $d(F') = \sum_{i=1}^{z-1} \left(a_i \sum_{j=1}^{i} x_j \right) + a_{z+1} \left(\sum_{j=1}^{z-1} x_j + x_{z+1} \right) + a_z (\sum_{j=1}^{z-1} x_j + x_{z+1} + x_z \Big) + \sum_{i=z+2}^{n} \left(a_i \sum_{j=1}^{i} x_j \right)$
- $d(F') - d(F)$
  $= \Bigg(a_z \left(\sum_{j=1}^{z-1} x_j + x_z \right) + a_{z+1} (\sum_{j=1}^{z-1} x_j + x_z + x_{z+1} \Big) \Bigg) - \Bigg(a_{z+1} \left(\sum_{j=1}^{z-1} x_j + x_{z+1} \right) + a_z (\sum_{j=1}^{z-1} x_j + x_{z+1} + x_z \Big) \Bigg)$

Bước 2: Xét hiệu $d(F') - d(F)$

Bước 2: Xét hiệu $d(F') - d(F)$

- $d(F') - d(F)$
$$= \left(a_z\left(\cancel{\sum_{j=1}^{z-1} x_j} + \cancel{x_z}\right) + a_{z+1}\left(\cancel{\sum_{j=1}^{z-1} x_j} + x_z + \cancel{x_{z+1}}\right)\right) -$$
$$\left(a_{z+1}\left(\cancel{\sum_{j=1}^{z-1} x_j} + \cancel{x_{z+1}}\right) + a_z\left(\cancel{\sum_{j=1}^{z-1} x_j} + x_{z+1} + \cancel{x_z}\right)\right)$$

Bước 2: Xét hiệu $d(F') - d(F)$

- $d(F') - d(F)$
$$= \left(a_z \left(\cancel{\sum_{j=1}^{z-1} x_j} + \cancel{x_z} \right) + a_{z+1} \left(\cancel{\sum_{j=1}^{z-1} x_j} + x_z + \cancel{x_{z+1}} \right) \right) -$$
$$\left(a_{z+1} \left(\cancel{\sum_{j=1}^{z-1} x_j} + \cancel{x_{z+1}} \right) + a_z \left(\cancel{\sum_{j=1}^{z-1} x_j} + x_{z+1} + \cancel{x_z} \right) \right)$$
- $d(F') - d(F) = a_{z+1} x_z - a_z x_{z+1}$

Bước 3: Tìm đặc điểm của O

- Mục tiêu là $d(X)$ càng nhỏ càng tốt, mà O thỏa mãn điều kiện ta cần tìm nên ta muốn $d(F') - d(F) \geq 0 \Rightarrow \boxed{a_{z+1}x_z - a_z x_{z+1} \geq 0} \Rightarrow \frac{a_z}{x_z} \leq \frac{a_{z+1}}{x_{z+1}}$

Bước 3: Tìm đặc điểm của O

- Mục tiêu là $d(X)$ càng nhỏ càng tốt, mà O thỏa mãn điều kiện ta cần tìm nên ta muốn $d(F') - d(F) \geq 0 \Rightarrow \boxed{a_{z+1}x_z - a_z x_{z+1} \geq 0} \Rightarrow \frac{a_z}{x_z} \leq \frac{a_{z+1}}{x_{z+1}}$
- Ta thấy điều kiện rất hợp lí. Nếu $\frac{a_z}{x_z}$ càng cao thì một lượt đánh càng có giá trị nên lượt đánh nào cao ta sẽ sử dụng trước.

Bước 3: Tìm đặc điểm của O

- Mục tiêu là $d(X)$ càng nhỏ càng tốt, mà O thỏa mãn điều kiện ta cần tìm nên ta muốn
  $d(F') - d(F) \geq 0 \Rightarrow \boxed{a_{z+1}x_z - a_z x_{z+1} \geq 0} \Rightarrow \frac{a_z}{x_z} \leq \frac{a_{z+1}}{x_{z+1}}$
- Ta thấy điều kiện rất hợp lí. Nếu $\frac{a_z}{x_z}$ càng cao thì một lượt đánh càng có giá trị nên lượt đánh nào cao ta sẽ sử dụng trước.
- Để chắc chắn, bạn làm thêm Bước 4: chứng minh điều kiện tham này đúng bằng Lập luận Tráo đổi.

Tower

Tower

Cho n hộp, hộp thứ i có khối lượng w_i, độ bền s_i và giá trị v_i
Tìm cách chọn k hộp trong số n hộp được cho để xây được ngọn tháp có tổng giá trị lớn nhất sao cho:

Tower

Cho n hộp, hộp thứ i có khối lượng w_i, độ bền s_i và giá trị v_i
Tìm cách chọn k hộp trong số n hộp được cho để xây được ngọn tháp có tổng giá trị lớn nhất sao cho:

- Các hộp được đặt chồng lên nhau.

Cho n hộp, hộp thứ i có khối lượng w_i, độ bền s_i và giá trị v_i
Tìm cách chọn k hộp trong số n hộp được cho để xây được ngọn tháp có tổng giá trị lớn nhất sao cho:

- Các hộp được đặt chồng lên nhau.
- Tổng khối lượng các hộp ở trên không lớn hơn độ bền của hộp ở dưới.

Cho n hộp, hộp thứ i có khối lượng w_i, độ bền s_i và giá trị v_i
Tìm cách chọn k hộp trong số n hộp được cho để xây được ngọn tháp có tổng giá trị lớn nhất sao cho:

- Các hộp được đặt chồng lên nhau.
- Tổng khối lượng các hộp ở trên không lớn hơn độ bền của hộp ở dưới.

Giới hạn: $n \leq 10^3, w_i, s_i \leq 10^4, v_i \leq 10^9$

Hướng giải

Hướng giải

- Đầu tiên, chọn các hộp để xây tháp từ nóc để dễ kiểm tra điều kiện.

Hướng giải

- Đầu tiên, chọn các hộp để xây tháp từ nóc để dễ kiểm tra điều kiện.
- Tạm thời bỏ qua giới hạn, ta thử nghĩ nếu làm bài này bằng Quy hoạch động thì ta sẽ cần lưu các thông tin gì:
    - Tập các hộp đã được xét. Có 2^n cách chọn tập này.
    - Tổng khối lượng các hộp đã được chọn. Lưu ý do không có hộp nào có độ bền w_i vượt quá 10^4, tổng khối lượng các hộp tối đa là 2×10^4.

Hướng giải

- Đầu tiên, chọn các hộp để xây tháp từ nóc để dễ kiểm tra điều kiện.
- Tạm thời bỏ qua giới hạn, ta thử nghĩ nếu làm bài này bằng Quy hoạch động thì ta sẽ cần lưu các thông tin gì:
  - Tập các hộp đã được xét. Có 2^n cách chọn tập này.
  - Tổng khối lượng các hộp đã được chọn. Lưu ý do không có hộp nào có độ bền w_i vượt quá 10^4, tổng khối lượng các hộp tối đa là 2×10^4.

  Tổng số trạng thái là $2^n \times 2 \times 10^4$, với $n \leq 10^3$ thì số trạng thái quá lớn.

Hướng giải

- Đầu tiên, chọn các hộp để xây tháp từ nóc để dễ kiểm tra điều kiện.
- Tạm thời bỏ qua giới hạn, ta thử nghĩ nếu làm bài này bằng Quy hoạch động thì ta sẽ cần lưu các thông tin gì:
    - Tập các hộp đã được xét. Có 2^n cách chọn tập này.
    - Tổng khối lượng các hộp đã được chọn. Lưu ý do không có hộp nào có độ bền w_i vượt quá 10^4, tổng khối lượng các hộp tối đa là 2×10^4.

  Tổng số trạng thái là $2^n \times 2 \times 10^4$, với $n \leq 10^3$ thì số trạng thái quá lớn.
- Ta tìm cách tối ưu để giảm số trạng thái. Sau khi thử nhiều cách khác nhau, ta thấy cách tốt nhất có lẽ là dùng Tham lam để sắp xếp các hộp theo thứ tự nào đó để đưa số trạng thái từ 2^n xuống còn n (tức tập các hộp đang xét sẽ chỉ là một hậu tố của một dãy nào đó).

Bước 1: Tìm $X, d(X)$

Bước 1: Tìm $X, d(X)$

- Biểu diễn X là một chỉnh hợp $x_1, x_2, ..., x_k$ chập k của n hộp. Các hộp xuất hiện trong X theo thứ tự từ trên xuống dưới trong ngọn tháp.

Bước 1: Tìm $X, d(X)$

- Biểu diễn X là một chỉnh hợp $x_1, x_2, ..., x_k$ chập k của n hộp. Các hộp xuất hiện trong X theo thứ tự từ trên xuống dưới trong ngọn tháp.
- $d(X) = 1$ nếu X là một cách chọn hợp lệ, $d(X) = 0$ nếu X không là cách chọn hợp lệ.

Bước 2: Giả sử tồn tại nghiệm F

Bước 2: Giả sử tồn tại nghiệm F

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn điều kiện gì đó mà ta chưa biết và ta có nghiệm F' thỏa mãn điều kiện hơn.

Bước 2: Giả sử tồn tại nghiệm F

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn điều kiện gì đó mà ta chưa biết và ta có nghiệm F' thỏa mãn điều kiện hơn. Giả sử điều kiện đó có liên quan tới hai hộp $z, z+1$, và ta chỉ cần tráo hai hộp này là F trở thành F'.

Bước 2: Giả sử tồn tại nghiệm F

Giả sử tồn tại nghiệm F tối ưu không thỏa mãn điều kiện gì đó mà ta chưa biết và ta có nghiệm F' thỏa mãn điều kiện hơn. Giả sử điều kiện đó có liên quan tới hai hộp $z, z+1$, và ta chỉ cần tráo hai hộp này là F trở thành F'. Đánh số lại các hộp theo thứ tự xuất hiện trong F, khi đó $F = 1, 2, ..., z, z+1, ..., k$ và $F' = 1, 2, ..., z+1, z, ..., k$.

Bước 2: Giả sử tồn tại nghiệm F

Bước 2: Giả sử tồn tại nghiệm F

- F thỏa mãn các điều kiện:
  - $s_2 \geq w_1$
  - $s_3 \geq w_1 + w_2$
  - ...
  - $s_z \geq \sum_{i=1}^{z-1} w_i$
  - $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z$
  - $s_{z+2} \geq \sum_{i=1}^{z+1} w_i$
  - ...
  - $s_k \geq \sum_{i=1}^{k-1} w_i$

Bước 2: Giả sử tồn tại nghiệm F

- F thỏa mãn các điều kiện:
  - $s_2 \geq w_1$
  - $s_3 \geq w_1 + w_2$
  - ...
  - $s_z \geq \sum_{i=1}^{z-1} w_i$
  - $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z$
  - $s_{z+2} \geq \sum_{i=1}^{z+1} w_i$
  - ...
  - $s_k \geq \sum_{i=1}^{k-1} w_i$
- F' cần thỏa mãn các điều kiện:
  - $s_2 \geq w_1$
  - $s_3 \geq w_1 + w_2$
  - ...
  - $s_{z+1} \geq \sum_{i=1}^{z-1} w_i$
  - $s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$
  - $s_{z+2} \geq \sum_{i=1}^{z+1} w_i$
  - ...
  - $s_k \geq \sum_{i=1}^{k-1} w_i$

Bước 3: Xác định đặc điểm O

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.
- Để $s_z \geq \sum_{i=1}^{z-1} w_i \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.
- Để $s_z \geq \sum_{i=1}^{z-1} w_i \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Ta có thể cần $w_{z+1} \leq 0$, nhưng điều này khó đáp ứng được.

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.
- Để $s_z \geq \sum_{i=1}^{z-1} w_i \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Ta có thể cần $w_{z+1} \leq 0$, nhưng điều này khó đáp ứng được.
- Để $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.
- Để $s_z \geq \sum_{i=1}^{z-1} w_i \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Ta có thể cần $w_{z+1} \leq 0$, nhưng điều này khó đáp ứng được.
- Để $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Phần tăng từ s_{z+1} lên $s_z \leq$ Phần tăng từ $\sum_{i=1}^{z-1} w_i + w_z$ lên $\sum_{i=1}^{z-1} w_i + w_{z+1}$

Bước 3: Xác định đặc điểm O

Làm thế nào để:

$$\begin{cases} s_z \geq \sum_{i=1}^{z-1} w_i \\ s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \end{cases} \Rightarrow \begin{cases} s_{z+1} \geq \sum_{i=1}^{z-1} w_i \\ s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1} \end{cases}$$

chắc chắn đúng?

- Ta thấy $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_{z+1} \geq \sum_{i=1}^{z-1} w_i$ đã chắc chắn đúng rồi.
- Để $s_z \geq \sum_{i=1}^{z-1} w_i \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Ta có thể cần $w_{z+1} \leq 0$, nhưng điều này khó đáp ứng được.
- Để $s_{z+1} \geq \sum_{i=1}^{z-1} w_i + w_z \Rightarrow s_z \geq \sum_{i=1}^{z-1} w_i + w_{z+1}$:
  - Phần tăng từ s_{z+1} lên $s_z \leq$ Phần tăng từ $\sum_{i=1}^{z-1} w_i + w_z$ lên $\sum_{i=1}^{z-1} w_i + w_{z+1}$
  - $s_z - s_{z+1} \leq w_{z+1} - w_z \Rightarrow \boxed{s_z + w_z \leq s_{z+1} + w_{z+1}}$

Bước 4

Bước 4

- Dùng Lập luận Tráo đổi, ta chứng minh nếu tồn tại nghiệm tối ưu F không thỏa mãn $s_z + w_z \leq s_{z+1} + w_{z+1}$ thì sẽ tồn tại nghiệm tối ưu O thỏa mãn điều kiện này.

Bước 4

- Dùng Lập luận Tráo đổi, ta chứng minh nếu tồn tại nghiệm tối ưu F không thỏa mãn $s_z + w_z \leq s_{z+1} + w_{z+1}$ thì sẽ tồn tại nghiệm tối ưu O thỏa mãn điều kiện này.
- Ta chỉ cần xét các nghiệm thỏa mãn $s_z + w_z \leq s_{z+1} + w_{z+1}$ trong lúc quy hoạch động $\Rightarrow$ thuật Quy hoạch động với độ phức tạp $O(nw)$