

PREVOI ONLINE TRAINNING 2022 - CONTEST #3

Teacher: Đỗ Phan Thuận
dophanthuan@gmail.com

Teaching Assitant: Nguyễn Đức Thuận
thuanbn03@gmail.com

Khoa Học Máy Tính
Đại học Bách Khoa Hà Nội



Ngày 11 tháng 10 năm 2022

Bài 1. Hạnh phúc

Bài 2. Chèo thuyền

Bài 3. Di chuyển trên ba hàng

Hạnh phúc — HAPPINESS

An và Bằng thay nhau trực N ca, mỗi ca phải có ít nhất một trong hai nhân viên trực. Với ca trực thứ i , ông An có chỉ số hạnh phúc A_i nếu trực và ông Bằng có chỉ số hạnh phúc B_i nếu trực vào ca đó. Hai ông đều rất hạnh phúc nếu như cả hai đều nhận được tổng chỉ số hạnh phúc không nhỏ hơn H .

Yêu cầu: Hãy tính xem có bao nhiêu cách phân công trực khác nhau sao cho cả hai đều thấy hạnh phúc.

Subtask 1 ($N \leq 12$): Duyệt toàn bộ $O(3^N)$

Đối với mỗi ca, ta có ba lựa chọn:

1. An trực một mình
2. Bằng trực một mình
3. Cả hai cùng trực

Subtask 2 ($N \leq 20$): Duyệt phân đôi tập $O(3^{N/2} \times N)$

Chia thành 2 tập, mỗi tập kích thước $[N/2]$, mỗi nhóm có cách phân công độc lập nhau. Duyệt toàn bộ trên mỗi tập. Với mỗi cặp điểm (h_1, h_2) duyệt được trên tập 1, tính số lượng cặp điểm (p_1, p_2) duyệt được trên tập 2 sao cho $p_1 + h_1 \geq H$ và $p_2 + h_2 \geq H$. Thuật toán chi tiết:

1. Lưu các cặp điểm duyệt được trên tập 1 vào A_1 , trên tập 2 vào A_2 .
2. Sắp xếp A_1 và A_2 theo giá trị đầu tiên rồi đến giá trị thứ hai của cặp giảm dần.
3. Lặp trên A_1 , với mỗi cặp (h_1, h_2) :
 - ▶ Tìm các cặp (p_1, p_2) trên A_2 với $p_1 \geq H - h_1$, đẩy p_2 vào cây phân đoạn
 - ▶ Tìm số lượng các phần tử $\geq H - h_2$ cộng dồn vào kết quả.

DPT: $O(3^{N/2} \times \log(3^{N/2})) \simeq O(3^{N/2} \times N)$.

Cách khác: **SOS DP (sum over subset)**: BTVN

Bài 1. Hạnh phúc

Bài 2. Chèo thuyền

Bài 3. Di chuyển trên ba hàng

Chèo thuyền — ROWBOAT

Có N cặp vợ chồng tổ chức đi chèo thuyền trong đó có tất cả M cặp vợ chồng mới cưới. $2N$ người ngồi xếp thành hàng dài từ trước ra sau.

Yêu cầu: Tính xem có bao nhiêu cách có thể để sắp xếp tất cả $2N$ người trên chiếc thuyền chèo, sao cho hai vợ chồng trong mỗi cặp trong số M cặp vợ chồng mới cưới không ngồi cạnh nhau?

Subtask 1 ($1 \leq M \leq N \leq 100$): QHĐ $O(N \times M)$

Gọi $f(x, y, z)$ là số cách sắp xếp thứ tự chỗ ngồi cho x người không thuộc các cặp mới cưới và y cặp vợ chồng mới cưới ($x + 2y$ người) sao cho:

- ▶ Không có cặp vợ chồng mới cưới nào mà hai người ngồi cạnh nhau;
- ▶ Nếu $z = 1$ nghĩa là có một người trong nhóm x người không được ngồi vào ghế đầu tiên còn trống; $z = 0$ nếu ngược lại.
- ▶ $f(0, 0, z) = 1$, nghĩa là đã xếp hết.

$f(x, y, 0)$ được tính như sau:

- ▶ xếp một người trong nhóm x người vào ghế đầu tiên. Số cách xếp trong trường hợp này là $x \times f(x - 1, y, 0)$;
- ▶ xếp một người trong nhóm y cặp vào ghế đầu tiên, người còn lại trong cặp không được xếp bên cạnh. Như vậy người này sẽ tính sang nhóm x người kia để thành $x + 1$ người. Số cách xếp trong trường hợp này là $2 \times y \times f(x + 1, y - 1, 1)$;
- ▶ $f(x, y, 0) = x \times f(x - 1, y, 0) + 2 \times y \times f(x + 1, y - 1, 1)$.

Tương tự:

$$f(x, y, 1) = (x - 1) \times f(x - 1, y, 0) + 2 \times y \times f(x + 1, y - 1, 0).$$

Subtask 2 ($1 \leq M \leq N \leq 10^5$):

Bao hàm loại trừ $O(N + M)$

- ▶ Gọi $g(k)$ là số lượng cách xếp chỗ sao cho với k cặp định trước, không có 2 người nào trong mỗi cặp là ngồi sát nhau.
- ▶ Số cách xếp chỗ cho k cặp này là $C(2N - k, k) \times k! \times 2^k$.
- ▶ Số lượng cách sắp xếp những người còn lại là $(2(N - k))!$.
- ▶ $g(k) = C(2N - k, k) \times k! \times 2^k \times (2(N - k))!$.
- ▶ Theo bao hàm loại trừ, kết quả bài toán là:
$$\sum (-1)^k \times g(k) \times C(M, k), \forall 0 \leq k \leq M.$$
- ▶ Sử dụng nghịch đảo mô-đun hoặc giản ước tử mẫu để tính kết quả chia lấy dư.
- ▶ BTVN: xếp trên vòng tròn.

Bài 1. Hạnh phúc

Bài 2. Chèo thuyền

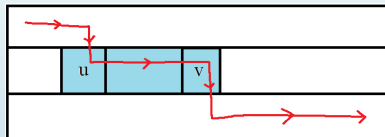
Bài 3. Di chuyển trên ba hàng

THREEROWS

- **Subtask 1 (15%):** $K_i = 0$ với $\forall i \in [1, Q]$.
- Mua hết tất cả các chìa khóa sau đó quy hoạch động trên lưới ô vuông giống bài toán tìm đường đi ngắn nhất.
- Gọi $dp[i, j]$ là điểm số đạt được lớn nhất khi đến ô (i, j) .
- ĐPT: $O(N)$ với hệ số lập trình là khoảng $c = 3$

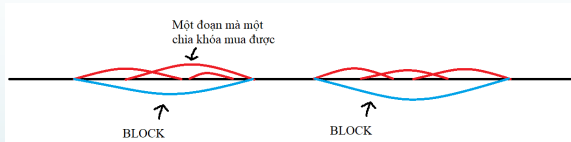
THREEROWS

- **Subtask 2 (15%):** $Q \leq 15$.
- Gọi $sum[i, j] = A_{i,1} + A_{i,2} + \dots + A_{i,j}$.
- Đặt:
 - $s[i] = sum[1, i] - sum[2, i - 1]$.
 - $f[i] = sum[2, i] + sum[3, n] - sum[3, i - 1]$.
- Di chuyển một đoạn $[u, v]$ trên hàng hai thì số điểm tối ưu bạn có được là $s[u] + f[v] - cost(u, v)$. Trong đó: $cost(u, v)$ là chi phí tối thiểu để mua các chìa khóa để mở hết các ô thuộc đoạn $[u, v]$ trên hàng 2.



THREEROWS

- Duyệt 2^Q cách mua chìa khóa.
- Mỗi cách sẽ phân hàng thứ hai thành các “block” liên tiếp gồm các ô được mở khóa.



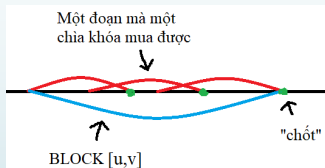
- Nếu đường đi nằm trọn trong block $[u, v]$, kết quả sẽ là $\max_{u \leq i \leq j \leq v} \{s[i] + f[j]\}$.
- Ta sẽ có tối đa Q block
- Với mỗi block $[u, v]$ ta sẽ tính giá trị $\max_{u \leq i \leq j \leq v} \{s[i] + f[j]\}$ bằng Segment Tree với độ phức tạp $O(\log N)$.
- ĐPT là $O(2^Q \cdot Q \cdot \log N)$.

THREEROWS

- **Subtask 3 (30%):** $N, Q \leq 300$.
- Dựa theo subtask 2 thì kết quả bài toán là $\max_{1 \leq i \leq j \leq N} \{s[i] + f[j] - cost(i, j)\}$.
- Gọi $dp[i] = \max_{1 \leq u \leq v \leq i} \{s[u] - cost(u, v)\}$.
- Duyệt qua hết Q loại chìa khóa, với chìa khóa loại (l, r, k) thỏa mãn $i \in [l, r]$ thì ta có:
$$dp[i] = \max_{l-1 \leq j < i} \{dp[j] - k\}.$$
- Kết quả bài toán là $\max_{1 \leq i \leq N} \{dp[i] + f[i]\}$.
- ĐPT: $O(N^2 \cdot Q)$.

THREEROWS

- **Subtask 4 (40%):** Không giới hạn gì thêm.
- Gọi các điểm R_i là “chốt”.
- Khi đi trên hàng thứ hai qua một block $[u,v]$, ta sẽ đi qua các “chốt”.
- Dựa vào những điểm “chốt” để quy hoạch động.



- Gọi $dp[i] = \max_{1 \leq u \leq v \leq i} \{s[u] - cost(u, v)\}$. Tuy nhiên ở bài toán này ta chỉ xét các $dp[i]$ với i là các điểm “chốt”.

THREEROWS

- Với một chìa khóa có dạng (L_i, R_i, K_i) ta sẽ cần lưu vào một vector $list[R_i].push(i)$
- Công thức quy hoạch động được xây dựng khi xét chìa khóa (l, r, k) thỏa mãn $r = i$:
 - ❑ $dp[i] = \max_{l \leq j \leq i} \{s[j] - k\}$ nếu i là điểm “chốt” đầu tiên khi xét từ trái qua phải.
 - ❑ Ngược lại, $dp[i] = \max_{l-1 \leq j < i} \{dp[j] - k\}$.

THREEROWS

- Kết quả bài toán sẽ được tính như sau nếu xem chìa khóa (l, r, k) là chìa khóa có r lớn nhất mà bạn mua (bên phải nhất):
 - ❑ $\max_{l \leq i \leq j \leq r} \{s[i] + f[j] - k\}$ nếu bạn chỉ mua một chìa khóa.
 - ❑ $\max_{l \leq i \leq j \leq r} \{dp[i] + f[j]\}$ nếu chìa khóa trước đó có điểm “chốt” nằm trong chìa khóa này.
 - ❑ $\max_{l \leq i \leq r} \{dp[l-1] + f[i] - k\}$ nếu chìa khóa trước đó có điểm “chốt” nằm ngoài chìa khóa này, cụ thể là ở vị trí $l-1$.
- Cải tiến toàn bộ lời giải trên bằng Segment Tree ta sẽ đạt được ĐPT $O(Q \cdot \log N)$.