

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

Dynamic programming p2

Báo cáo bài tập CS112.N21.KHTN

Giảng viên: Th.S Nguyễn Thanh Sơn

Họ và tên	MSSV
Trần Hoàng Bảo Ly	21521109
Lê Thu Hà	21520800

Hồ Chí Minh, tháng 6 năm 2023

Mục lục

1. Bài 1: Repost.	3
1.1. Đề bài:	3
1.2. Nhận xét:	3
2. Bài 2: Glass.....	4
2.1. Đề bài.....	4
2.2. Nhận xét.....	5
3. Bài 3: Elavator.....	6
3.1. Đề bài.....	6
3.2. Nhận xét.....	6

Yêu cầu chung: Chỉ ra đâu là optimal substructure và overlapping subproblems trong 3 bài toán mà nhóm đăng trên wecode.

1. Bài 1: Repost.

1.1. Đề bài:

Một ngày nọ, Polycarp đăng một bức ảnh trên mạng xã hội Facebook. Nhiều người bạn của anh ấy chia sẻ lại bức ảnh, một số khác thì trực tiếp đăng lại bức ảnh đó. Việc đăng và chia sẻ ảnh là các chuỗi liên nhau có dạng "tên1 reposted tên2", trong đó tên1 là tên của người đã chia sẻ bức ảnh và tên2 là tên của người đã đăng bức ảnh. Biết rằng, vào thời điểm chia sẻ ảnh, với mỗi chuỗi "tên1 reposted tên2" thì "tên1" chưa có bức ảnh trong bảng tin của họ và "tên2" đã có ảnh trong bảng tin của họ. Polycarp đã được đăng ký là "Polycarp" và ban đầu bức ảnh chỉ có trong bảng tin của anh ấy. Tìm mức độ phổ biến của bức ảnh mà Polycarp đã đăng, biết mức độ phổ biến của bức ảnh là độ dài của chuỗi chia sẻ ảnh lớn nhất.

Input

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 200$) là số lần chia sẻ.
- n dòng sau là các chuỗi chia sẻ ảnh theo thứ tự. Mỗi chuỗi trong số chúng được viết trên một dòng duy nhất có dạng "tên1 reposted tên2". Tất cả các tên trong đầu vào có thể bao gồm chữ cái viết thường hoặc viết hoa, chữ số có độ dài từ 2 đến 24 ký tự. Tên người dùng không phân biệt chữ hoa chữ thường.

Output

Tìm mức độ phổ biến bức ảnh (độ dài chuỗi chia sẻ ảnh)

1.2. Nhận xét:

Bài toán dưới đây là bài toán tìm chuỗi chia sẻ dài nhất bắt đầu từ một người, ta không quan tâm chuỗi đó như thế nào, ta chỉ quan tâm chuỗi dài nhất bắt đầu từ người đó. Với bài toán trên hai tính chất optimal structure và overlapping subproblem lần lượt là:

Optimal Substructure (cấu trúc con tối ưu): Khi tính toán độ dài chuỗi chia sẻ ảnh tại mỗi bước, ta cần xác định độ dài chuỗi chia sẻ ảnh lớn nhất có thể bắt đầu từ một

người dùng cụ thể. Từ đó, ta có thể xây dựng một giải pháp tối ưu toàn cục bằng cách so sánh các giải pháp tối ưu cục bộ tại mỗi bước.

Overlapping Subproblem (Bài toán con gối nhau): Khi tính toán độ dài chuỗi chia sẻ ảnh tại mỗi bước, ta có thể gặp lại các người dùng đã được tính toán trước đó. Do đó, ta có thể lưu trữ các giá trị đã tính để tránh tính toán lại chúng. Điều này giúp tối ưu hóa thời gian tính toán bằng cách tránh tính toán trùng lặp.

2. Bài 2: Glass

2.1. Đề bài

Nhân ngày quốc tế thiếu nhi, mẹ cho Thương đi trung tâm thương mại chơi. Trong trung tâm ngày này có một dãy các gian hàng giới thiệu máy làm Mixue. Thương rất thích trà sữa nên kéo mẹ vào một gian hàng. Tại gian hàng này có kê một cái bàn, trên bàn có để máy làm Mixue và n cốc trà sữa Mixue đã được làm sẵn. Thương rất muốn uống tất cả trà sữa ở trong các cốc, nhưng cô lại xấu hổ nên không muốn uống nhiều hơn k cốc. Rất may, Thương có thể đổ nước từ cốc này sang cốc khác. Vấn đề của Thương bây giờ là không biết chọn cốc nào, vì khoảng cách của tất cả các cốc đến cô đều như nhau. Biết rằng, để đổ nước từ cốc i sang cốc j thì mất chi phí là C_{ij} . Bạn hãy giúp Thương xác định thứ tự đổ nước từ công này sang cốc khác sao cho tổng chi phí là nhỏ nhất có thể.

Input

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq k \leq n \leq 20$).
- n dòng tiếp theo, mỗi dòng chứa n số nguyên C_{ij} ($0 \leq C_{ij} \leq 105$). Trong đó, số ở dòng i và cột j có giá trị C_{ij} và $C_{ii} = 0$.

Output

Gồm duy nhất 1 dòng ghi tổng chi phí nhỏ nhất có thể.

Sample Input 1

```
5 2
0 5 4 3 2
7 0 4 4 4
3 3 0 1 2
```

4 3 1 0 5

4 5 5 5 0

Sample Output 1

5

Giải thích:

- Đầu tiên Thương đổ trà sữa từ cốc 4 sang cốc 3 mất chi phí là 1
- Sau đó Thương đổ trà sữa từ cốc 3 sang cốc 5 mất chi phí là 2
- Cuối cùng Thương đổ trà sữa từ cốc 1 sang cốc 5 mất chi phí là 2
- Như vậy tổng chi phí tối thiểu là: $1+2+2=5$

Note: Có 40% số test có $n \leq 10$

2.2. Nhận xét

Sử dụng dynamic programming with bitmask để giải quyết. Giá trị 0 nghĩa là cốc nước chưa đổ sang cốc khác và giá trị 1 nghĩa là cốc nước đã được đổ sang cốc khác.

Ở mỗi bước ta tiến hành xét lần lượt các cốc U có mask là 1 và cốc V có mask là 0: $dp[mask]$ (mask ở trạng thái hiện tại) = $\min(dp[mask], dp[mask \& \sim(1 \ll u)] + a[u][v])$. Nghĩa là giữ nguyên trạng thái hiện tại, hoặc quay về trạng thái cốc u chưa bị đổ sang cốc khác (mask = 0) và đổ cốc v sang cốc u nếu như chi phí thấp hơn trạng thái hiện tại.

Optimal substructure (cấu trúc con tối ưu): Trong trường hợp này, ta có thể thấy rằng tại mỗi bước, việc chọn cốc tiếp theo để đổ nước vào sẽ phụ thuộc vào thứ tự các cốc trước đó đã được chọn. Vì vậy, việc tìm cách chọn cốc để đổ nước sao cho tổng chi phí là nhỏ nhất có thể có tính chất optimal substructure.

Overlapping subproblem (bài toán con gối nhau): Ta cũng có thể nhận thấy rằng việc tính toán chi phí cho các bước con sẽ có sự gối vào nhau. Khi chúng ta đổ nước từ cốc u sang cốc v , chi phí này có thể được tính dựa trên việc đổ nước từ cốc u đến các cốc khác trước đó. Do đó, ta có tính chất overlapping subproblem trong bài toán này.

3. Bài 3: Elevator

3.1. Đề bài

Có n người ở tầng G muốn đi lên tầng cao nhất của tòa nhà bằng thang máy trong khi tòa nhà đó chỉ có 1 thang máy. Biết khối lượng của mỗi người và tải trọng của thang máy. Bạn hãy tìm ra số chuyến thang máy tối thiểu để đưa n người lên tầng cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n và x ($1 \leq n \leq 20$, $1 \leq x \leq 109$) lần lượt là số người và sức tải tối đa của thang máy (kg).
- Dòng thứ hai chứa n số nguyên dương $w_1, w_2, w_3, \dots, w_n$ ($1 \leq w_i \leq x$): khối lượng của mỗi người (kg).

Output

Gồm duy nhất 1 dòng ghi số chuyến thang máy tối thiểu.

Sample Input 1

4 10

4 8 6 1

Sample Output 1

2

Giải thích:

- Chuyến đầu tiên, thang máy chở người 1 và người 3: $4 + 6 = 10 \leq 10$
- Chuyến thứ hai, thang máy chở người 2 và người 4: $8 + 1 = 9 \leq 10$

3.2. Nhận xét

Sử dụng dynamic programming with bitmask để giải quyết. Giá trị 1 nghĩa là người đó đã lên tầng cao nhất và ngược lại. Mảng dp lưu hai giá trị:

Giá trị thứ nhất là số chuyến ít nhất với trạng thái mask đang xét.

Giá trị thứ hai là khối lượng ít nhất của chuyến cuối cùng đối với trạng thái mask hiện tại đang xét.

Ở mỗi bước ta tiến hành đưa người thứ i vào thang máy nếu như khối lượng của người đó bé hơn khối lượng có thể chở còn lại của thang máy ở trạng thái người thứ i chưa được vào thang máy.

$dp[mask] = \min(dp[mask], \text{make_pair}(dp[mask \ \& \ \sim(1 \ll i)].\text{first}, dp[mask \ \& \ \sim(1 \ll i)].\text{second} + a[i]));$

Ngược lại đời chuyển mới

$dp[mask] = \min(dp[mask], \text{make_pair}(dp[mask \ \& \ \sim(1 \ll i)].\text{first} + 1, a[i]));$

Trong trường hợp này, ta có thể thấy rằng tại mỗi bước, việc đưa người thứ i vào chuyên thang máy hiện tại hay chuyển mới phụ thuộc vào các trạng trước đó (Bài toán con gói nhau). Và nếu $dp[mask]$ hiện tại ở trạng thái tối ưu thì $dp[mask]$ ở các trạng thái tiếp theo sẽ thu được kết quả tối ưu nên bài toán này cũng có tính chất (cấu trúc con tối ưu).