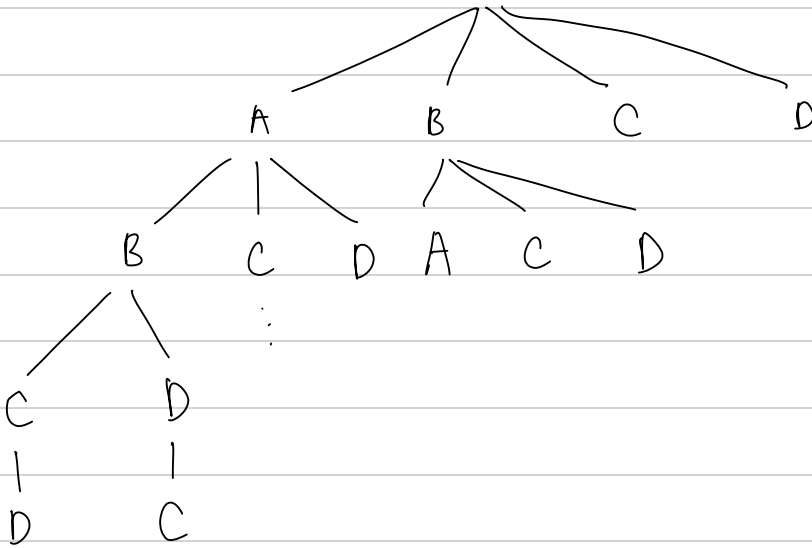


Thực chất bài toán là biến thể của liệt kê xâu nhị phân

↳ Vẽ đc cây cho trường idx và value \Rightarrow OK!

vd1 Bài toán TSP (đường đi không trùng lặp)



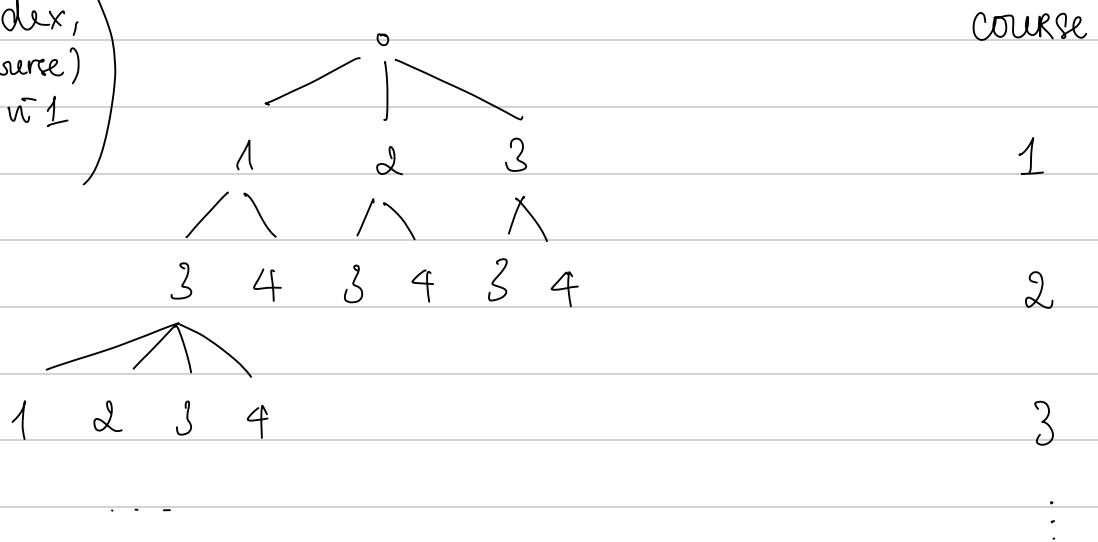
② BCA (Bài toán tối ưu, duyệt qua nhánh cần)

- Cố gắng nhìn thấy \approx bin sequence : Có nhìn ra CÂY ko?
- Giảng viên & course : cái nào là idx?

\Rightarrow Course i^{th} : $i \rightarrow \text{index}$ (vì 1 Course chỉ có thể đợc dạy bởi 1 teacher)

Teacher j^{th} : $j \rightarrow \text{value}$

(nếu teacher = index,
node 0 cây(course)
sẽ là set thay vì 1
gtại đợc mề!)



↳ Xử lí mâu thuẫn ở ngoài (course conflict)
tránh đợc vào phép duyệt

12

stdin		stdout
	4 12	3
1	5 1 3 5 10 12	
2	5 9 3 4 8 12	
3	6 1 2 3 4 9 7	
4	7 1 2 3 5 6 10 11	
	5	
	12	
	13	
	15	
	24	
	25	

TEACHER

COURSE

COURSE
CONFLICT

* Cách xây cây

→ Có nhiều cách để xây đợc cây khi có constraint

- (1) Loại bỏ gtại ở tnh ở ngoài, rồi mới đi vào vòng lặp
- (2) Loại bỏ "số" ở ngoài, vào vòng lặp \rightarrow loại bỏ chỉ hết
- (3) Xử lí mâu thuẫn bên trong vòng lặp, tận dụng quá trình

duyệt

```
for v in [1...n]
  check(...)
```

+) Xâu cây chỉ là visualize \emptyset gian n_0 , chứ \emptyset phải xây cây thật

*) Tối ưu minimal load

+) Duyệt tổng thể chứ \emptyset phải heuristic \rightarrow luôn đảm bảo tìm đc

+) Ý tưởng: tìm min của max

\uparrow maxload giữa tất cả load của các gv

\hookrightarrow next iter: nếu load mình assign cho 1 gv $>$ maxload
 \rightarrow biết là \emptyset phải tối ưu \rightarrow loại bỏ luôn!

⑧ CBUS

+) Đọc kĩ constraint: $1 \leq n \leq 11 \rightarrow$ xâu 22 \Rightarrow backtracking

+) Biểu diễn 1 n_0 :

KHOẢNG $\xrightarrow{i+n}$ ĐIỂM ĐẾN
 $1...n$ $1+n, 2+n, \dots 2n$

	o	o	o	...	o	o	o	o	...	o
	1				n	n+1	n+2			2n
	11				11					
(chưa đón)	0				0	(chưa trả)				
(đã đón)	1				1	(đã trả)				

000...000

\downarrow

\vdots

111...111

\rightarrow nếu $\theta = 1 \Rightarrow$ ghi nhận n_0

Buổi thực hành số 2

Bài toán 1. Đếm số nghiệm nguyên

Đếm số nghiệm nguyên của phương trình $a_1X_1 + a_2X_2 + \dots + a_nX_n = M$

- Given 2 positive integers n, M and n positive integers a_1, a_2, \dots, a_n . Compute the number of positive integer solutions to the equation: $a_1X_1 + a_2X_2 + \dots + a_nX_n = M$
- Input**
 - Dòng 1: n và M
 - Dòng 2: a_1, a_2, \dots, a_n
- Output**
 - Số nghiệm nguyên dương

stdin	stdout
3 5 1 1 1	6

Gợi ý: (Complete search)

- Thuật toán vét cạn, vét hết khả năng có thể
- Giá trị x_i có thể nhận được sẽ từ 0 tới M/a_i
- Thuật toán để vét cạn?
 - Dùng vòng lặp
 - Dùng đệ quy quay lui

Câu hỏi chuẩn bị

Câu hỏi 1.

Trong bài này nếu dùng đệ quy sẽ có nhược điểm gì?

Câu hỏi 2.

Độ phức tạp tính toán theo O-lớn trong trường hợp tồi nhất sẽ là bao nhiêu? Đánh giá theo n và M

Câu hỏi 3.

Có cách gì tính được số lượng nghiệm mà KHÔNG cần vét cạn?

Bài toán 2. CBUS

- Có n hành khách $1, 2, \dots, n$. hành khách thứ i muốn di chuyển từ điểm i tới điểm $i+n$ ($n=1,2,\dots,n$).
- Có 1 xe bus đỗ ở điểm 0 và có k vị trí ghế ngồi.
- Bạn có 1 ma trận khoảng cách C , trong đó $c(i,j)$ là khoảng cách từ điểm i tới j ($i, j = 0, 1, \dots, 2n$)
- Tính toán hành trình ngắn nhất cho xe bus để có thể phục vụ n hành khách và quay trở về điểm đỗ 0
- **Đầu vào**
- Dòng 1 là giá trị của n và k ($1 \leq n \leq 11, 1 \leq k \leq 10$)
- Các dòng tiếp theo là dòng của ma trận (rows and columns are indexed from 0, 1, 2, ..., $2n$).
- **Đầu ra**
- Tổng quãng đường của hành trình ngắn nhất

stdin	stdout
3 2 0 8 5 1 10 5 9 9 0 5 6 6 2 8 2 2 0 3 8 7 2 5 3 4 0 3 2 7 9 6 8 7 0 9 10 3 8 10 6 5 0 2 3 4 4 5 2 2 0	25

Gợi ý

- Vét cạn các cách di chuyển? Không gian vét cạn lớn?
- Dùng back tracking? Vẫn phải dò hết các nhánh
- Hạn chế bớt các nhánh bằng việc thêm 1 hàm bound (cận dưới) để tỉa bớt số lượng nhánh phải dò
- Hàm bound lựa chọn thế nào?
 - Theo tổng quãng đường tới vị trí hiện tại?
 - Hàm bound phức tạp hơn?
 - Liệu có thể làm tương tự thuật toán dijkstra trong tìm đường đi ngắn nhất?

Câu hỏi chuẩn bị

Câu 1. Giới hạn dưới của hành trình trong ví dụ minh họa là bao nhiêu nếu xe chỉ chở 1 người?

Câu 2. Nếu xe chỉ được chở 1 người thì thuật toán vét cạn sẽ có độ phức tạp cỡ nào?

Câu hỏi 3. Thuật toán nhánh-và-cận trong trường hợp tồi nhất có tốt hơn 1 thuật toán vét cạn thông thường?

Câu hỏi 4. Nếu xe chỉ chở 1 người tới đích sau đó mới đón người tiếp thì với ví dụ minh họa ở trên, tổng quãng đường nhỏ nhất sẽ là?

Bài toán 3. Balanced Course Assignment (BCA)

- At the beginning of the semester, the head of a computer science department D have to assign courses to teachers in a balanced way. The department D has m teachers $T = \{1, 2, \dots, m\}$ and n courses $C = \{1, 2, \dots, n\}$. Each teacher $t \in T$ has a preference list which is a list of courses he/she can teach depending on his/her specialization. We

known a list of pairs of conflicting two courses that cannot be assigned to the same teacher as these courses have been already scheduled in the same slot of the timetable. The load of a teacher is the number of courses assigned to her/him. How to assign n courses to m teacher such that each course assigned to a teacher is in his/her preference list, no two conflicting courses are assigned to the same teacher, and the maximal load is minimal.

- Input

- The input consists of following lines
- Line 1: contains two integer m and n ($1 \leq m \leq 10$, $1 \leq n \leq 30$)
- Line $i+1$: contains an positive integer k and k positive integers indicating the courses that teacher i can teach ($\forall i=1, \dots, m$)
- Line $m+2$: contains an integer k
- Line $i+m+2$: contains two integer i and j indicating two conflicting courses ($\forall i=1, \dots, k$)

- Output

- The output contains a unique number which is the maximal load of the teachers in the solution found and the value -1 if not solution found.

stdin	stdout
4 12 5 1 3 5 10 12 5 9 3 4 8 12 6 1 2 3 4 9 7 7 1 2 3 5 6 10 11 5 1 2 1 3 1 5 2 4 2 5	3

Gợi ý.

- Cách chia dựa trên vết cận số khả năng có thể sẽ là cách chia n phần tử thành m nhóm khác nhau
- Dùng back tracking để thử chọn cách phân cho 1 giáo viên mà không vi phạm ràng buộc (tương tự sudoku hoặc bài toán xếp hậu)
- Yêu cầu số lượng môn dạy lớn nhất của 1 giáo viên trong kỳ càng nhỏ càng tốt? kết hợp thêm bound trong back tracking ở trên để ưu tiên hướng duyệt mà có số môn phân cho 1 giáo viên nhỏ nhất có thể

số course số gv

- Bạn có thể phân công lần lượt mỗi giáo viên 1 môn, sau đó lại quay vòng phân công tiếp thêm môn thứ 2

Câu hỏi chuẩn bị

Câu 1. Có bao nhiêu cách chia n phần tử thành m nhóm (các phần tử trong mỗi nhóm không cần có thứ tự)? Stirling nb of the 2nd kind $\therefore S(n, m)$

$n < m \Rightarrow S(n, m) = 0$ $n = m \Rightarrow S(n, m) = 1$	$n > m$ $S(n, m) = m * S(n-1, m) + S(n-1, m-1)$
--------------------------------------------------------------------	----------------------------------------------------

↓
m choices to place 1 element \rightarrow 1 group

Câu 2. Cận dưới của việc chia n việc cho m người là bao nhiêu trong trường hợp KHÔNG ràng buộc? Nếu có ràng buộc thì việc gán người 1 cho công việc 1 sẽ dẫn đến cận chia việc của những người còn lại như thế nào?

Câu 3. Trong ví dụ minh họa nếu người 1 chọn môn 1 và người 2 chọn môn 3 thì những người còn lại còn bao nhiêu lựa chọn có thể?

đều mất đi 2 lựa chọn (1 và 3).