Mecánicos y Averías

PID: 904

Contexto

Estás trabajando de informático/a en una empresa que se dedica a reparar maquinaria pesada. Cada mecánico puede reparar ciertos tipos de avería. La reparación de una avería requiere de la asignación de un mecánico, y cada mecánico puede estar asignado cada día a un máximo de una avería. La empresa te pide que diseñes un programa que permita optimizar la asignación diaria de mecánicos a averías, de modo que se consiga reparar el máximo posible de averías cada día.

El Problema

Tenemos M mecánicos y A averías por reparar. Una tabla bidimensional C, de elementos booleanos $c(i,j)$, indica que el mecánico i tiene capacidad para reparar la avería j.

Se debe realizar una asignación de mecánicos a averías que maximice el número de averías reparadas en el día, teniendo en cuenta que cada mecánico puede ser asignado a una única avería.

Entrada

La primera línea de la entrada contiene un entero, P, que indica el número de casos de prueba.

Cada caso de prueba contiene $1+M$ líneas:

- la primera línea tiene dos enteros, M y A, que indican el número de mecánicos y de averías, respectivamente.
- las siguientes M líneas contienen T enteros cada una, separados por espacios en blanco, correspondiendo el entero j-ésimo de la línea i-ésima la capacidad del mecánico i-ésimo para reparar la avería j-ésima, es decir, el elemento c(i,j) de la tabla C.

Salida

La primera línea de la salida contiene un entero, P, que indica el número de resultados.

Para cada caso resultado, la salida deberá tener dos líneas:

- la primera línea es el número total de averías reparadas.
- la segunda línea indica qué mecánico se asignó a cada avería. Contendrá A números enteros, separados por espacios en blanco, donde el entero j-ésimo indica el mecánico asignado a la avería j-ésima; para las averías no reparadas se utilizará el número 0.

En caso de haber varias posibles soluciones óptimas, deberás elegir aquella que en la salida se codifique con el menor número. Por ejemplo, en el segundo ejemplo de abajo son válidas las soluciones "1 2 4", "2 4 1" y "4 2 1", luego la solución será la primera.

No obstante, los ceros deben dejarse al final. Dicho de otro modo, en el árbol de backtracking se debe generar primero 1, 2, 3, ... y por último el 0. Por ejemplo, si una solución es "1 2 4 0 0" y la otra es "0 0 1 2 4" o bien "1 0 2 0 4", se elegirá la primera.

Ejemplo de Entrada

```
3
2 3
0 1 0
0 0 0
```

```
4 3
1 0 1
1 1 0
0 0 0
1 1 1
2 3
0 0 1
1 0 1
```

Ejemplo de Salida

```
3
1
0 1 0
3
1 2 4
2
2 0 1
```

Ejemplos Extendidos

Entrada Salida

Joaquín Cervera López, 2008
"Algoritmos y Estructuras de Datos"
Departamento de Informática y Sistemas - Universidad de Murcia