Conjuntos Parcialmente Ordenados.

Teorema de Dilworth

# Definiciones

## Conjunto Parcialmente Ordenado (Partially Ordered Set):

Sea *S* un conjunto de elementos y  ≤  un orden parcial. O sea, para algunos elementos de x,y de *S* se tiene *x* ≤ *y*, y debe ser:

* Reflexiva (*x* ≤ *x*)
* Antisimétrica (si *x* ≤ *y* y *y* ≤ *x*, entonces *x* = *y*)
* Transitiva (si *x* ≤ *y* y *y* ≤ *z*, entonces *x* ≤ *z*)

Un ejemplo de conjunto parcialmente ordenado es el de los puntos (*x*, *y*) en el plano, con el operador (*x*1, *y*1) ≤ (*x*2, *y*2) si y solo si *x*1 ≤ *x*2 y *y*1 ≤ *y*2 (ejemplo problema G NCPC 2007).

## Cadena:

Se define una **cadena** *C* como un subconjunto de *S* {*x*1, *x*2, …, *xn* } tal que *x*1 ≤ *x*2 ≤ … ≤ *xn*.

## Anticadena:

Se define una **anticadena** *A* como un subconjunto de *S* tal que para todo par de elementos *x*,*y* de *A* no se cumple *x* ≤ *y* ni *y* ≤ *x*. O sea en una anticadena no existe un par de elementos que sean comparables. Al tamaño de la mayor anticadena se le llama ancho del conjunto y se va a denotar como **a(S)**.

## Partición:

Una **particion** *P* es un conjunto de cadenas de forma tal que cada elemento de S pertenece a exactamente una cadena. Al tamaño de la menor partición se le va a denotar como **mp(P)**.

# Teorena de Dilworth

El teorema plantea que el tamaño de la **mayor anticadena** en un conjunto finito parcialmente ordenado S es igual al tamaño de la **menor partición**, o sea que **a(S) = mp(S)**.

## Demostración

Se va a hacer inducción fuerte en el tamaño del conjunto. Para el conjunto vacío se cumple, ya que el tamaño de la mayor anticadena es 0 que es igual al tamaño de la menor partición.

Sea P un conjunto finito parcialmente ordenado con |P| > 0.

Es evidente que a(P) ≤ mp(P), ya que si se tiene una partición con mp(P) cadenas, cualquier anticadena con mp(P)+1 o más elementos debería tener dos o más elementos en una misma cadena de la partición, contradiciendo que dichos elementos no estén relacionados.

A continuación se va a demostrar que a(P) ≥ mp(P), o sea, que existe una partición en a lo sumo a(P) cadenas. Sea C una cadena de tamaño máximo posible en S.

Si a(P\C) < a(P), entonces por hipótesis de inducción P\C se puede descomponer en a(P\C) cadenas, al agregar la cadena C se obtiene una partición en a(P\C)+1 cadenas, pero a(P\C)+1 ≤ a(P), que es lo que se quería demostrar.

Si a(P\C) = a(P). Entonces se va a tomar una anticadena A en P\C tal que |A| = a(P) y se van a considerar los conjuntos:

A- = {x∈P: ∃y ∈A: x ≤y}

A+ = {x∈P: ∃y ∈A: y ≤x}

Por definición de A- y A+, se tiene que A⊆A- y A⊆ A+. Sea m el máximo elemento de C, se va a probar que C no pertenece a A-.

Efectivamente como C es una cadena maximal, no existe y tal que m ≤ y, m ≠ y, así que para que m pertenezca a A-, m debe pertenecer a A, pero A ∩ C = ∅, ya que A es una anticadena de P\C.

Por lo tanto | A- | < |P| y como A- tiene una anticadena de tamaño a(P) que es A (no puede tener una mayor ya que A- ⊆ P) entonces por hipótesis de inducción A- se puede descomponer en ciertas cadenas C1−,…, Ca(P)−.

Se va a probar que A ∩ Ci− posee un único elemento y es el máximo de Ci-. La intersección de una cadena con una anticadena posee a lo sumo un elemento, pero A tiene a(P) elementos, se tiene una partición de A- en a(P) cadenas, y A ⊆ A-, así que cada elemento de A está en una cadena diferente, y por lo tanto hay exactamente un elemento de A en cada cadena. Sea x el elemento de ci que pertenece a A, se va a probar que es el máximo de Ci−. Si no lo fuera, sea mi el máximo de Ci, como Ci− ⊆ A- existe un y ∈ A: mi ≤ y. Entonces se tendría que x ≤ mi ≤ y, es decir x e y elementos distintos relacionados de una anticadena, lo cual es una contradicción.

Haciendo lo propio con A+, es decir mín C ∉ A+ implica que | A+| < |P| y por hipótesis de inducción entonces A+ se puede descomponer en las cadenas C1+,…,Ca(P)+, tales que mín Ci+ ∈ A.

Sin pérdida de generalidad se pueden reordenar los índices y asumir que máx Ci- = mín Ci+. Se va a probar que C1,…,Ca(P) definidas como Ci = Ci- ∪ Ci+ es una partición de P con a(P) cadenas.

Si x, y ∈ Ci, entonces:

* x, y ∈ Ci-, lo cual implica x,y están relacionados.
* x, y ∈ Ci+, y entonces x,y están relacionados.
* x ∈ Ci-, y ∈ Ci+, x ≤ mi ≤ y y entonces x ≤y.

Además son disjuntas pues si x ∈ Ci ∩ Cj entonces:

* x ∈ Ci+ ∩ Cj- = ∅.
* x ∈ Ci+ ∩ Cj+ = ∅.
* x ∈ Ci- ∩ Cj+, lo cual implica que mj ≤ x ≤ mi, pero mj y mi pertenecen a la anticadena A.
* x ∈ Ci+ ∩ Cj-, lo cual implica que mi ≤ x ≤ mj, pero mj y mi pertenecen a la anticadena A.

Por lo tanto existe una partición de P en a(P) cadenas así que mp(P) ≤ a(P) que era lo que se quería demostrar.

## Calculando la mayor anticadena

Por el teorema de Dilworth, el tamaño de la mayor anticadena es igual al tamaño de la menor partición en cadenas. A continuación se va a probar como hallar la menor partición a través de la construcción de un grafo bipartito y de un emparejamiento máximo en el mismo.

Sea G = <V,U,E> un grafo bipartito con V = P, U = P. Sea vx nodo correspondiente al elemento x de P en V, y de forma análoga se define ux. Se define además E = { (vx,uy), x,y ∈ P con x ¡= y, x<=y}.

A continuación se va a probar que una descomposición en m cadenas de P es equivalente a un emparejamiento de n-m aristas en G.

Sean c1,c2,…,cm las cadenas de una partición. Para cada cadena ci sus elementos se pueden ordenar de forma tal que: ci1<=ci2<=… cik, con |ci| = k, y para cada j = 1,…,k-1 se va a añadir la arista (vcij,ucij) al emparejamiento. El emparejamiento es válido ya que se puede verificar que cada nodo pertenece a lo sumo a una arista. Como hay m cadenas y por cada cadena c se añaden |c|-1 aristas al emparejamiento, se tiene un emparejamiento de tamaño |c1|-1+|c2|-1+…+|cm|-1 = n-m.

A continuación se va a probar que un emparejamiento de m aristas en G es equivalente a una descomposición de P en n-m cadenas.

Sea M un emparejamiento en G de m aristas. Inicialmente se considera una partición de n cadenas, donde cada cadena está formada por un único elemento. Para cada arista (ux,vy) ∈ M, se van a unir las cadenas correspondientes a x y y. Esto se puede hacer para toda arista de E, ya que en el momento en el que se toma una arista (ux, vy), como esta es la única arista a la que pertenece ux, x debe ser el elemento máximo de su cadena, y y el elemento mínimo de la suya, y como x <= y las cadenas se pueden unir. Cada vez que se unen dos cadenas se reduce en uno la cantidad total, así que como el emparejamiento tiene m aristas se va a obtener una descomposición de n-m cadenas.

Por lo tanto hallar la menor descomposición en cadenas de P es equivalente al mayor emparejamiento que se puede obtener en G.

# Dual del teorema de Dilworth

El teorema dual de Dilworth plantea que el tamaño de la mayor cadena un conjunto parcialmente ordenado es igual a la menor cantidad de anticadenas en que se puede particionar.

## Demostración

Para cada elemento x sea N(x) el tamaño de la mayor anticadena que termina en él. Sea T(i) el conjunto de todos los elementos x tales que N(x) = i. A lo sumo habrá L conjuntos T(i), donde L es el tamaño de la mayor anticadena, y para todo i los elementos que pertenecen a T(i) forman una anticadena, de lo contrario, si hubiera dos elementos x,y en T(i) tales que x <= y, s.p.g., la cadena de x se pudiera alargar adicionando y, y entonces y no perteneciera a T(i). Por lo tanto existe una partición en anticadenas con a lo sumo L elementos.

Pero cualquier partición en anticadenas tiene que tener al menos L elementos, ya que cada uno de los nodos de la cadena de tamaño L, tienen que estar en conjuntos diferentes.

# Problemas

## Problem G NCPC 2007 - Nested Dolls

Dilworth is the world’s most prominent collector of Russian nested dolls: he literally has thousands of them! You know, the wooden hollow dolls of different sizes of which the smallest doll is contained in the second smallest, and this doll is in turn contained in the next one and so forth. One day he wonders if there is another way of nesting them so he will end up with fewer nested dolls? After all, that would make his collection even more magnificent! He unpacks each nested doll and measures the width and height of each contained doll. A doll with width w1 and height h1 will fit in another doll of width w2 and height h2 if and only if w1 < w2 and h1< h2. Can you help him calculate the smallest number of nested dolls possible to assemble from his massive list of measurements?

#### Entrada

On the first line of input is a single positive integer 1 ≤ t ≤ 20 specifying the number of test cases to follow. Each test case begins with a positive integer 1 ≤ m ≤ 20000 on a line of itself telling the number of dolls in the test case. Next follow 2m positive integers w1 , h1, w2 , h2, . . . , wm, hm, where wi is the width and hi is the height of doll number i. 1 ≤ wi , h i ≤ 10000 for all i.

#### Salida

For each test case there should be one line of output containing the minimum number of nested dolls possible.

#### Solución

Para cada muñeca se conoce su ancho y su largo, así que podemos representarlas como (h,w). El conjunto de las muñecas es un conjunto parcialmente ordenado considerando la operación <=, donde (h1,w1) <= (h2,w2) si h1 <= h2 y w1 <= w2. Nótese que en el resultado final cada muñeca es una cadena de muñecas originales, luego el problema es equivalente a hallar la menor cantidad de cadenas en que se puede particionar el conjunto. Pero por el teorena de Dilworth el tamaño de la menor partición es igual al tamaño de la mayor anticadena.

Para entender mejor como hallar la mayor anticadena se consideran las muñecas como puntos en el plano ((h,w)=>(x,y)).

Para cada punto se va a calcular la mayor anticadena que termina en él. La mayor anticadena del conjunto tiene que terminar en algún punto, así que recorriendo todos los puntos se obtendría la solución. Esto se puede hacer mediante programación dinámica. Para cada punto solo hay que considerar los puntos que están a su izquierda y por encima de él. Se busca el punto que tanga la mayor anticadena y se le suma uno. La solución queda O(N\*N).

Se puede obtener una mejor solución con una estructura que permita hallar máximo en un rango en O(LogN). De esta forma la solución quedaría O(N\*logN).

1

1

2

3

4

2

5

3

6

4

M

Ejemplo:

Para la muñeca M solo hay que analizar los puntos en el área sombreada. La solución es 6.

#### Otra solución

Se recorren los puntos ordenados por x. Se mantiene un conjunto de pares (a,b) que indican que hay una anticadena de tamaño b que termina en un punto con valor a de y. Por lo tanto para un punto (x,y) se puede insertar (y,b+1) siempre que y < a.

Si se tienen dos elementos en el conjunto (a,b) y (c,d) tales que c<=a y d <=b entonces el último es redundante y se puede eliminar. De esta forma el conjunto permanece pequeño y solo hay que chequear un solo elemento para insertar, que es (a,b) con el mínimo valor de a tal que a >= y. Todo esto se puede hacer con un set en C++.

## MOG - Campañas publicitarias

Un país contiene N ciudades conectadas por carreteras unidireccionales. Si hay una carretera que va de A a B entonces se puede decir que hay un camino que va de A a B. Si hay un camino que va de A a B y otro camino que va de B a C, entonces se puede decir que hay un camino de A a C.

Un conjunto de ciudades se dice que es ***"Viable"*** si para todo par de ciudades A,B que pertenecen al conjunto se cumple exactamente una de las dos condiciones siguientes:

* **Hay un camino que va de A a B y otro camino que va de B a A** (los caminos pudieran involucrar ciudades que no estén en el grupo. (1)
* **No hay camino que vaya de A a B y no hay camino que vaya de B a A.** (2)

Algunas marcas están haciendo campañas publicitarias en el país. Cada marca puede hacer su campaña en solo un conjunto de ciudades, y este debe ser ***“Viable”***. El presidente decide que no quiere que haya muchas marcas, para que el país no se vuelva un caos publicitario, y que cada ciudad debe participar en exactamente una campaña. Determine cuál es la menor cantidad de campañas que se pueden hacer.

#### Entrada

En la primera línea se especificarán dos enteros separados por un espacio 1 <= N, E <= 100000, la cantidad de ciudades y la cantidad de carreteras.

A continuación habrá E líneas con dos enteros A y B separados por un espacio indicando que hay una carretera de la ciudad A a la ciudad B. Las ciudades están numeradas de 0 a N-1.

#### Salida

Entero M que sea la menor cantidad de marcas que pueden hacer su campaña en el país.

#### Solución

Se tiene un grafo dirgido. Nótese que si se tiene una componente fuertemente conexa todas las ciudades involucaradas cumplen la condición 1, o sea, una componente fuertemente conexa es un conjunto ***“Viable”***. Pero un conjunto viable puede estar compuesto por varias componentes fuertemente conexas, de forma tal que no exista ningún camino entre dos componentes.

Sea S el grafo formado por las componentes fuertemente conexas, entonces S es un grafo dirigido acíclico. Si se considera la operación ***hay camino*** entre dos nodos del grafo, entonces se tiene un conjunto parcialmente ordenado. Se puede probar que el problema es equivalente a hallar la menor cantidad de anticadenas en que se puede particionar este conjunto. Pero por el teorema Dual de Dilworth, esto es equivalente a hallar la mayor cadena en el conjunto, que es lo mismo que hallar el camino más largo del grafo. Como el grafo es un DAG el tamaño de esta camino se puede hallar fácilmente utilizando un orden topológico del mismo.

Para hallar el grafo de las componentes fuertemente conexas se necesita O(V+E) y para hallar el camino más largo se necesita O(V+E), por lo que la solución sería O(V+E).

## COJ 2288 - BIA

Byteland Information Agency has a network with N computers, each of these numbered from 1 to N. These computers are connected by unidirectional wires, through which information is transmitted. At one point, the special agent in charge of sending messages between multiple-computer performs several operations. An operation start with:

The agent select a computer from the agency that has not been selected before and send a message to all computers which are connected with selected station. You can safely assume that each computer automatic forwards the message to computers that are connected to it. Note that messages can't be sent in the opposite direction of the wire.

You has been hired by the agency to solve the following problem. They want to know what is the maximum number of computers which can send one message and they never receive any message (included messages which start in other station). Please, try to do your job quickly, you just have four hours to do it.

#### Entrada

The first line contains an integer T representing the number of test cases. The following T lines contains an integer N (1 <= N <= 50) representing the number of computers in the network. Then, a matrix M with NxN characters 'N' or 'Y'. if M[i][j] = 'Y', it means that there is a link between computer i and computer j, otherwise, there is no link between both computers.

#### Salida

The output contains T lines. Each line contains the required answer for each case.

#### Solución

El problema pide hallar el mayor conjunto de computadoras que se pudieron haber seleccionado, de forma tal que no recibieron ningún mensaje. Si se tiene una componente fuertemente conexa que contenga más de un nodo, cualquier computadora que envíe un mensaje lo va a recibir, por lo tanto, la solución tiene que ser un conjunto de computadoras que pertenezcan a componentes fuertemente conexas de un solo nodo.

Se halla el grafo de las componentes fuertemente conexas y su clausura, o sea se establecen aristas entre todo par de nodos entre los cuales exista un camino, pero se eliminan aquellas componentes que contengan más de un nodo. Ahora se tiene un conjunto parcialmente ordenado respecto a la relación ***hay camino***, y se quiere hallar el tamaño de la mayor anticadena. Pero el tamaño de la mayor anticadena es igual al tamaño de la menor partición en cadenas, y esto se puede calcular mediante el emparejamiento antes mencionado.