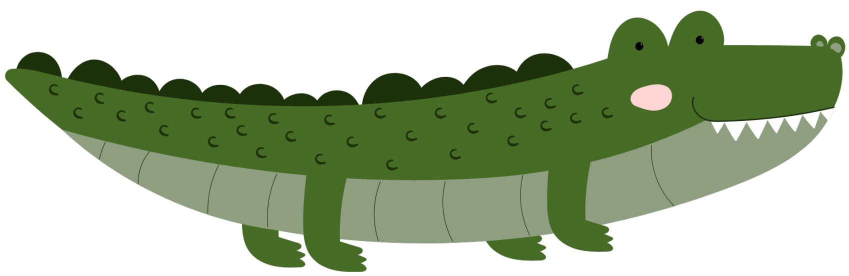
Olimpiada Informática Femenina III
Final día 1
`cocodrilos`

Lágrimas de cocodrilo

En una fiesta de disfraces de cocodrilos, cada cocodrilo lleva un cartel con un nombre. Como los cocodrilos son muy orgullosos no quieren que se sepa que están llorando. Aún así, sabemos que un cocodrilo llora si el nombre de su cartel tiene exactamente dos vocales diferentes (aunque pueden aparecer más veces en la palabra).
Por ejemplo ”gorila” tiene 3 vocales, por lo tanto no lloraría. En cambio ”cocodrilo” tiene 2 vocales diferentes, o sea que sí lloraría.
¿Cuántos de los cocodrilos de la fiesta lloran?



Entrada

Una línea con un entero T: el número de casos de entrada. Para cada caso de entrada:

- Una línea con un entero n: el número de cocodrilos. n líneas, cada una con una palabra (solo letras minúsculas, sin acentos), representando la palabra de cada cocodrilo.

Restricciones

- $1 \leq T \leq 50$
- $1 \leq n \leq 10^5$
- Las palabras nunca tendrán más de 10 letras.
- 50 puntos: las palabras tienen exactamente 4 letras y las vocales se encuentran en la segunda y cuarta letra.
- 50 puntos: restricciones originales

Salida

Una línea por caso de entrada con un único entero con el número de cocodrilos que están llorando.

Ejemplo

Entrada	Salida
1 10 rata elefante cocodrilo chimpance gorila murcielago zorro gallina cisne ballena	5

Hay un único caso de entrada. De estas palabras: rata y zorro solo tienen un tipo de vocal, elefante, cocodrilo, gallina, cisne y ballena cumplen la restricción de tener dos. El resto de palabras tiene más de dos tipos de vocal.

Goleada histórica

Hace unos meses que España cayó ante Marruecos en octavos de final del mundial, algo que dejó tristes a muchos aficionados, ya que llevaban cuatro años esperando el torneo tras ser eliminados en 2018. Y no solo esto, sino que España empezó el torneo goleando siete a cero a Costa Rica, dando todo un recital.

Tras este arranque en el mundial, fueron muchos los aficionados que empezaron a creerse que podíamos ganar el campeonato. Y es que no era para menos. Goleadas así son muy poco comunes en el fútbol. Blanca y Jacobo, que tienen una gran pasión por el fútbol y la programación, han decidido analizar datos históricos de fútbol para ver qué goleadas de este calibre se han producido a lo largo de los años en las distintas competiciones de este deporte.

Han decidido hacer un ránking, donde se ordene cada partido por cómo de difícil es conseguir el resultado obtenido para el ganador de cada partido. Y es que en esto entran muchos factores: si un equipo juega o no en casa, cuánta presión hay del público etc.

Han acordado la siguiente métrica: si el ganador de un partido ha metido g coles y el perdedor p goles, la puntuación será al menos $g - p$. Pero claro, si el ganador jugaba fuera de casa, ha tenido un plus de complejidad el partido, por lo que hay que sumar 1 a este resultado. Además, si el partido era en una competición internacional, multiplicaremos el resultado (incluyendo ese posible punto extra) por 2. Con esta métrica, ¿cuál de los partidos en la base de datos de Blanca y Jacobo se merece el reconocimiento como la mayor goleada?

Entrada

Una línea con un entero n: el número de partidos en la base de datos, seguida de n líneas (una por partido), cada una con el siguiente formato:

- El nombre de la competición en la que se produjo el partido. Este será `mundial`, `champions` o `liga`. Las dos primeras de estas tres competiciones son internacionales, mientras que la tercera es de ámbito nacional.
- El nombre del equipo ganador (sin espacios)
- El número de goles g del equipo ganador
- El nombre del equipo perdedor (sin espacios)
- El número de goles c del equipo perdedor
- Dónde se jugó el partido (`casa` o `fuera` en función de dónde jugó el ganador)

Restricciones

- $1 \leq n \leq 10^5$
- $0 \leq g, c \leq 100$
- Los nombres de los equipos serán cadenas de letras minúsculas del alfabeto inglés de no más de 30 caracteres.
- Ningún partido ha terminado en empate.
- Habrá un único partido con la mejor puntuación de la base de datos.
- 33 puntos: todos los partidos fueron de liga y con el vencedor jugando en casa.
- 67 puntos: restricciones originales.

Salida

Una línea con el siguiente formato:
La mayor goleada de la base de datos es del [INSERTA EQUIPO GANADOR] en [INSERTA COMPETICIÓN] con una diferencia de [INSERTA LA DIFERENCIA DE GOLES] goles.

Ejemplo

Entrada	Salida

`3`
`liga barcelona 2 madrid 1 fuera`
`mundial alemania 2 francia 0 casa`
`champions madrid 5 liverpool 2 fuera`
	`La mayor goleada de la base de datos es del madrid en champions con una diferencia de 3 goles.`

Tenemos las siguientes puntuaciones:

1. El primer partido lo ganó el `barcelona` fuera de casa, con una diferencia de $2 - 1 = 1$ gol. Al ser fuera, sumamos 1 a este total, dando una puntuación de 2. Como el partido fue en una competición nacional, este es el resultado final.

2. El segundo partido lo ganó `alemania` en casa, con una diferencia de $2 - 0 = 2$ goles. Al ser en casa, no sumamos 1, pero como fue en una competición internacional, multiplicamos esta diferencia por dos, dando una puntuación final de 4.

3. El tercer partido lo ganó el `madrid` fuera de casa, con una diferencia de $5 - 2 = 3$ goles. Al ser fuera, sumamos 1, dando un total de 4, que duplicamos para obtener la puntuación final de 8 por ser un torneo internacional.

Por lo tanto, la mayor puntuación es la del `madrid` en `champions` y esto será lo que pongamos en la salida.

Globos de Up

En la película Up atan un montón de globos a una casa para poder volar. Nosotros queremos hacer lo mismo. Ya tenemos n globos y sabemos los kilos que puede levantar cada globo: a_1 kilos el primero, a_2 kilos el segundo, y así sucesivamente hasta a_n kilos el enésimo. Ahora nos falta comprar las cuerdas. Las cuerdas cuestan más en función de la tensión que tienen que soportar: si una cuerda sujeta un globo que puede levantar x kg va a costar x euros y si sujeta un grupo de globos que en total pueden levantar y kg, la cuerda cuesta y euros.

Las cuerdas son muy cortas y, por lo tanto, tenemos que juntarlas con nudos. En los nudos se juntan los extremos de al menos tres cuerdas distintas. Cada extremo puede puede estar en un nudo, sujetando un solo globo o atado a la casa. Solo podemos atar un extremo de una cuerda a la casa.

En función del número de nudos que queramos hacer vamos a tener que comprar un número distinto de cuerdas. Si queremos hacer un solo nudo, tendremos que comprar $n+1$ cuerdas, ya que vamos a unir cada globo con el nudo usando una cuerda y luego usaremos una cuerda extra para conectar el nudo con la casa.

Podemos ver que el número de nudos que vamos a hacer está entre 1 y $n-1$. Para cada uno de estos valores debes calcular el precio mínimo que pueden costar las cuerdas para conseguir atar todos los globos a la casa con ese número exacto de nudos.

Entrada

Una línea con un entero n y otra línea con n enteros: $a_1, a_2, \ldots, a_n$.

Restricciones

- $2 \leq n \leq 10^5$
- $1 \leq a_i \leq 10^8$ para todo i
- 7 puntos: $a_i = 1$ para todo i
- 8 puntos: $a_i = k$ para todo i
- 10 puntos: $a_i = i$ para todo i
- 35 puntos: $n \leq 100$
- 40 puntos: sin restricciones adicionales

Salida

Una línea para cada número de nudos (de 1 a $n-1$) con el precio mínimo en euros que podemos conseguir.

Ejemplos

Entrada	Salida
3 1 2 3	12 15

Actualizando el sistema

Jacobo, después de estar una semana enfermo, tuvo una epifanía. Decidió, tras años de hacer cosas que le apasionaban a medias, crear una empresa que fabrica sistemas informáticos: OINet. Más concretamente, el objeto de la empresa es diseñar redes de aparatos electrónicos y cableado para gigantes de la industria. Dedica muchas horas cada día (y a veces por las noches) a pensar en cómo mejorar sus sistemas. Y claro, esto significa que cada poco hay una nueva, y mejor, versión de su producto. Sus clientes (k empresas) no cambian sus sistemas tan rápido como avanzan las actualizaciones de Jacobo, pero de vez en cuando vienen y compran la versión más nueva.

Con tanta nueva actualización, a Jacobo le vendría bien un programa que le ayudara a calcular cómo de buena es la versión actual comparada con alguna versión del pasado. ¿Sabrías ayudarle?

Más concretamente, todas las versiones del sistema de Jacobo tienen los mismos n aparatos y $n-1$ cables conectándolos. Entre una versión y otra lo que cambian son los tiempos de trayecto de los cables (cuánto tardan los datos en atravesarlos), pero no qué dos dispositivos conecta cada cable (bidireccional). En el sistema hay, para cada pareja de dispositivos x e y, exactamente una manera de mandar datos de x a y (con este trayecto posiblemente pasando a través de otros dispositivos). Todas las empresas empiezan adquiriendo la primera versión del programa y será la que tengan hasta que se te indique lo contrario.

Tu programa ha de procesar q consultas de varios tipos:

1. La empresa x ha actualizado sus sistemas a la última versión
2. ¿Cuánto tardan los datos en viajar del dispositivo i al dispositivo j en la versión del sistema que tiene actualmente la empresa z?
3. Jacobo ha lanzado una nueva versión del sistema. Cambia el tiempo de trayecto del cable que conecta los dispositivos x e y a t segundos.

Entrada

Una línea con tres enteros: n, k y q.
$n-1$ líneas, cada una con tres enteros: x_i, y_i y t_i, los dispositivos que conecta el i-ésimo cable y su tiempo de trayecto en la primera versión del producto.
q líneas, cada una con una consulta. Estas tienen los siguientes formatos, en función de su tipo (indicado por el primer entero de la línea, como puedes ver):

1. 1 x
2. 2 i j z
3. 3 x y t

Todos los números están indexados empezando por 0, incluyendo las versiones del sistema, las empresas, los cables, los dispositivos, etc.

Restricciones

- $1 \leq k, n, q \leq 10^5$
- $2 \leq n \leq 10^5$
- Los tiempos de los cables (los t_i y sus valores en futuras versiones) nunca superarán 10^4
- 11 puntos: $1 \leq k, n, q \leq 20$
- 7 puntos: todos los tiempos de trayecto serán iguales a 1 y todas las consultas serán de tipo 2
- 14 puntos: todas las consultas serán de tipo 2
- 15 puntos: $k = 1$
- 17 puntos: en las consultas de tipo 2, $i = 0$ siempre
- 36 puntos: restricciones originales

Salida

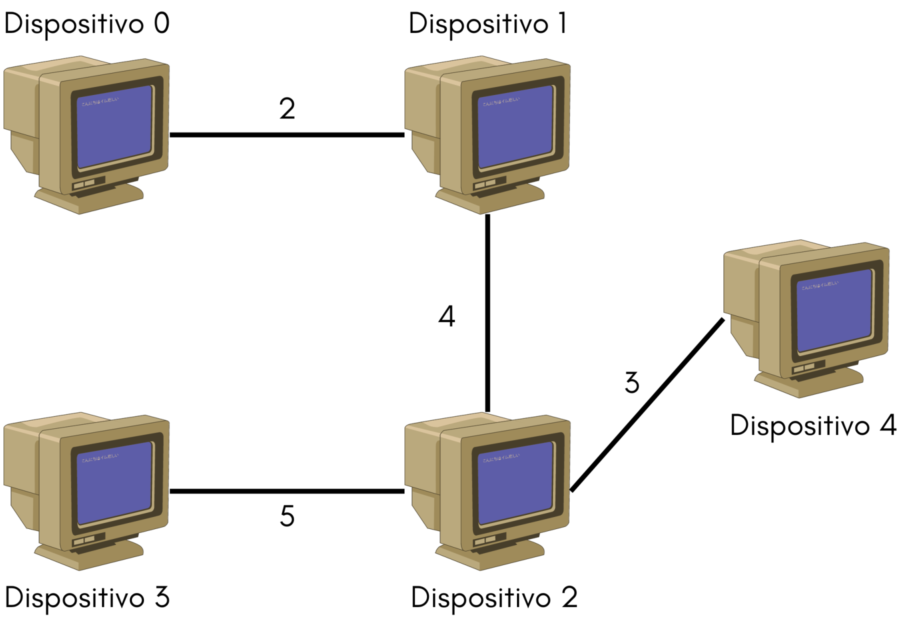
Una línea por cada consulta de tipo 2 con el resultado esperado.

Ejemplo

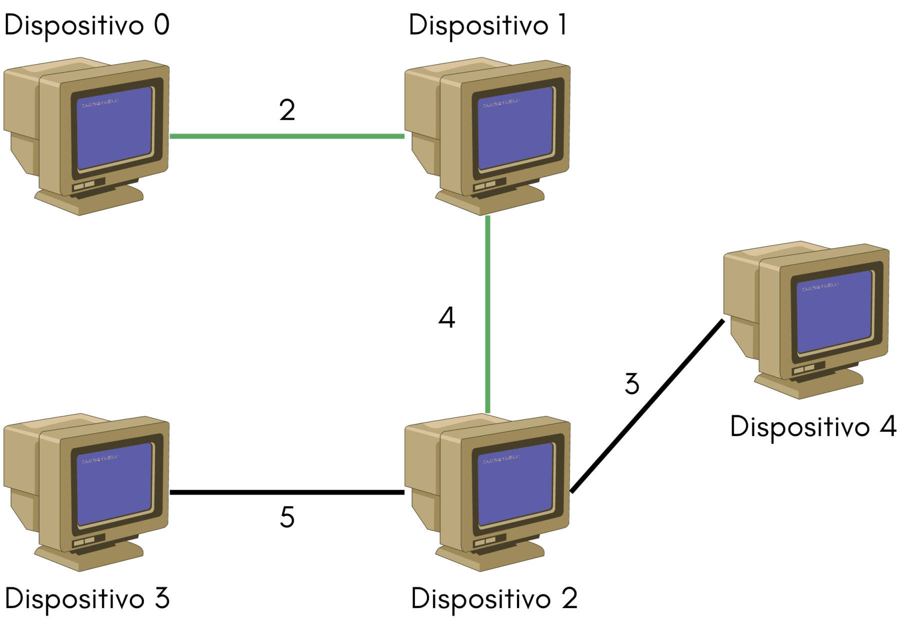
Entrada	Salida
5 2 6 0 1 2 1 2 4 2 3 5 2 4 3 2 0 2 0 2 3 4 1 3 0 1 8 1 1 2 0 2 1 2 0 2 0	6 8 12 6

[explicación en la página siguiente]

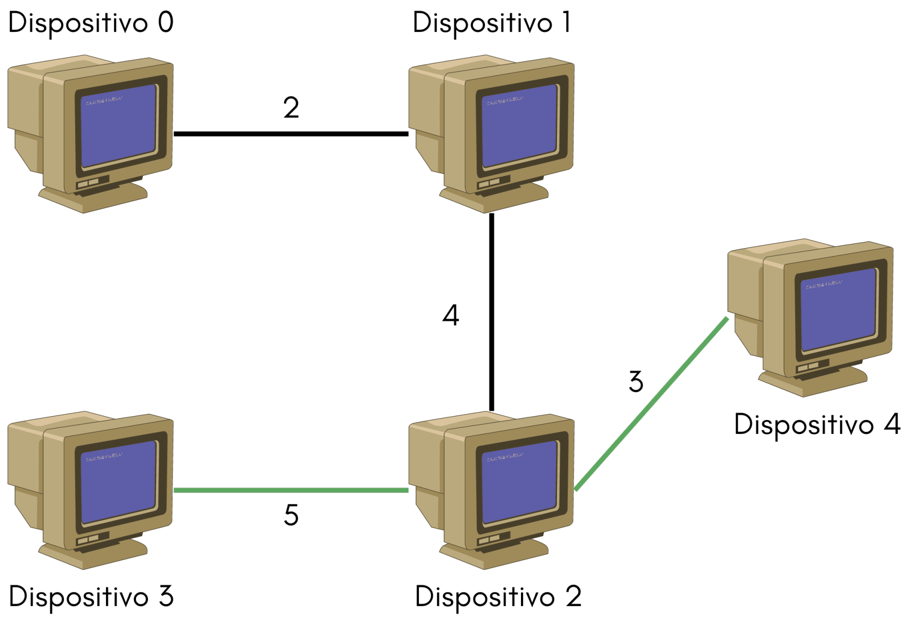
El sistema inicial de Jacobo es este:



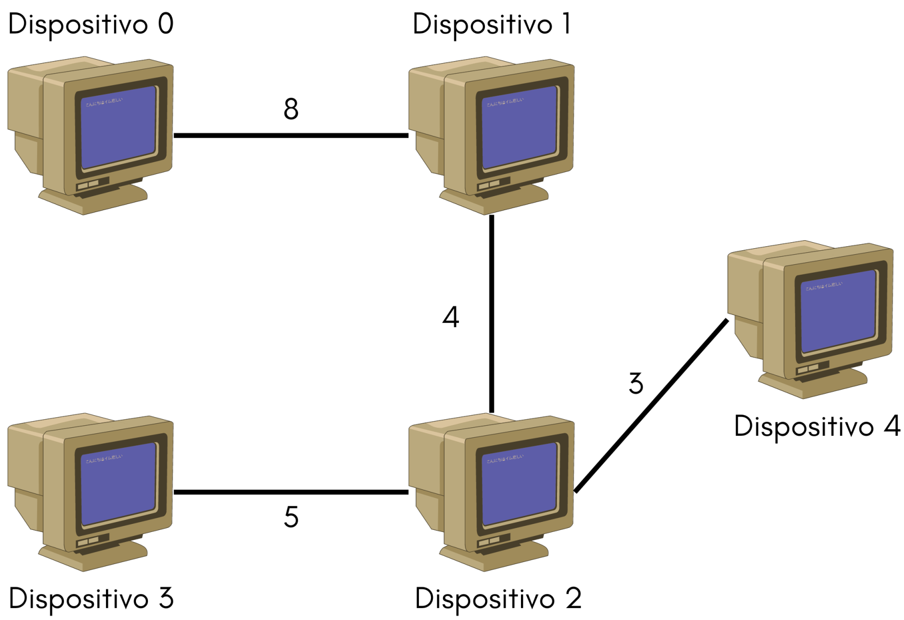
Ambas empresas tienen la primera versión del producto, y podemos calcular el tiempo que tardan los datos en llegar del ordenador 0 al 2 de la siguiente manera:



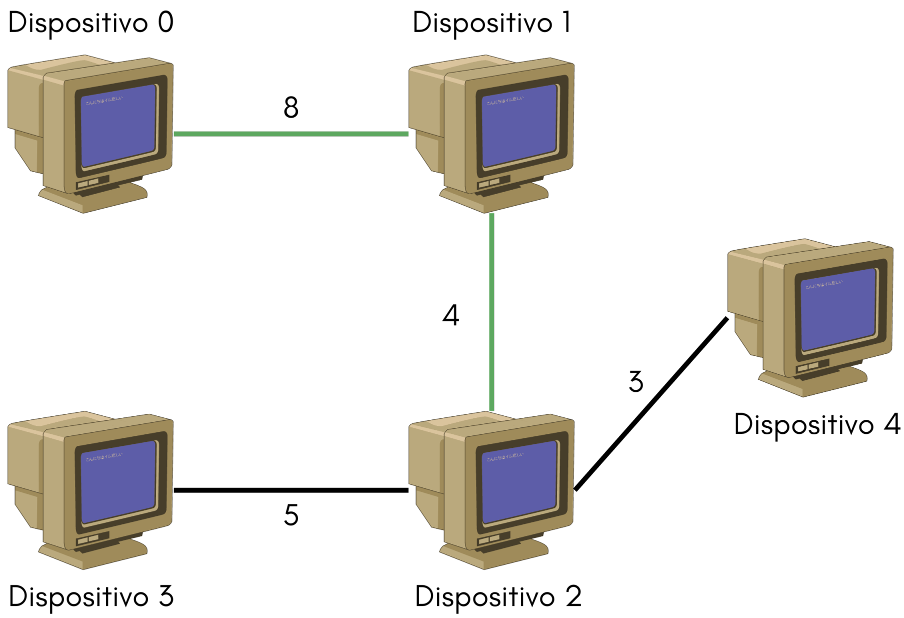
Ambas empresas tienen la primera versión del producto, y podemos calcular el tiempo que tardan los datos en llegar del ordenador 3 al 4 de la siguiente manera:



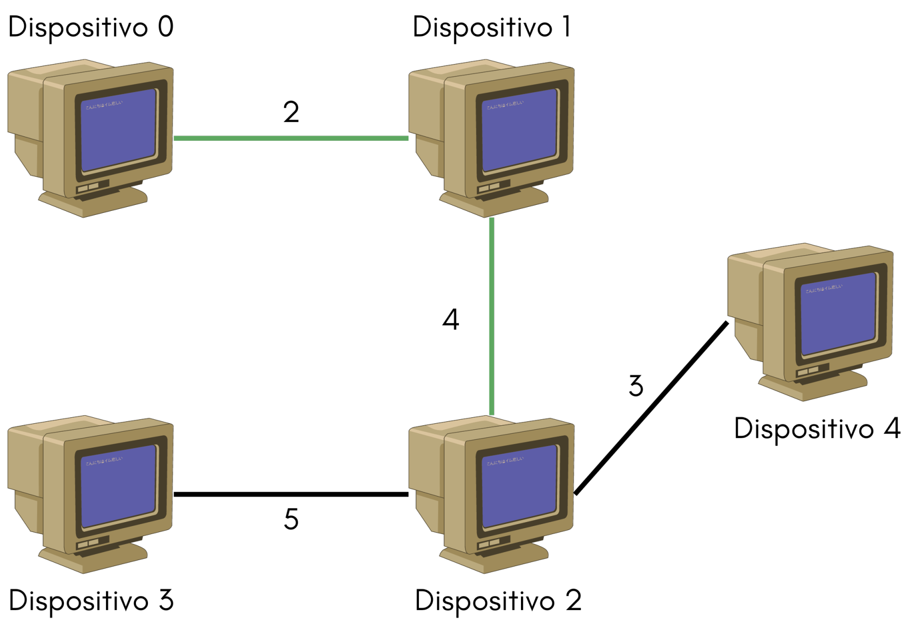
Jacobo diseña una nueva versión del sistema y ahora los datos tardan algo más en llegar del ordenador 3 al 4:



La empresa 1 actualiza su sistema a esta nueva versión, mientras que la 0 mantiene su sistema intacto. Por lo tanto, podemos calcular el tiempo que tardan en llegar los datos desde el ordenador 0 al 2 en el sistema de la empresa 1 de la siguiente forma:



y en el sistema de la empresa 0 de la siguiente forma:



N en raya

Joana y Lucía están jugando a un juego nuevo: el n *en raya*. El juego consiste en una cuadrícula de tamaño $n \times n$. En cada casilla hay que dibujar una cruz o un círculo. Por turnos cada jugador elige una casilla vacía y dibuja uno de los símbolos (el que quiera). El juego se acabará cuando todas las casillas estén llenas.

Una vez las $n \times n$ casillas están llenas hay que decidir quién ha ganado. Si existe alguna fila o columna en la que todas las casillas tienen el mismo símbolo, Joana ha ganado. En caso contrario, gana Lucía.

Dados varios valores de n, Lucía quiere saber, para cada n, de cuántas formas diferentes puede quedar la cuadrícula $n \times n$ al final del juego de manera que ella gane.

Entrada

Una línea con un entero t: el número de casos de entrada.

La siguiente línea contiene t enteros n (un valor distinto de n para cada caso de entrada).

Restricciones

$1 \leq t \leq 100$

- 13 puntos: $1 \leq n \leq 6$
- 23 puntos: $1 \leq n \leq 100$
- 23 puntos: $1 \leq n \leq 1000$
- 41 puntos: $1 \leq n \leq 10^4$

Salida

Una línea para cada valor de n_i con el número de cuadrículas finales de $n_i \times n_i$ en las que gana Lucía.
El número puede ser muy grande- escribe la respuesta módulo $10^9 + 7$.

Ejemplos

Entrada	Salida
1 10000	582696560
3 1 2 3	0 2 102

Multas

Entre finales de octubre y principios de noviembre se celebran las fiestas de Girona: Sant Narcís, el patrón de la ciudad. El día antes, se celebra desde hace más de diez años el bicicrucis. Como su numbre indica, esta actividad es una especie de viacrucis en bicicleta. Se organiza de manera popular y el ayuntamiento no tiene nada que ver; de hecho, suele haber tantas bicicletas de golpe que se bloquean algunas calles del centro de la ciudad.

La actividad es muy sencilla, una bicicleta va delante con un altavoz haciendo de guía y a lo largo del recorrido se hacen p paradas. En cada una de las paradas hay servicio de barra y música, y esta suele consistir en una neverita y un altavoz.

Últimamente la gente está empezando a usar la bicicleta pública de la ciudad, la Girocleta, para el bicicrucis. La Girocleta es un servicio público que consiste en diferentes estaciones donde se pueden aparcar las bicicletas y recogerlas con una tarjeta. El problema es que hay un tiempo máximo de uso de t minutos y, pasados estos t minutos, te cobran extra por el uso de la bicicleta. Para evitar estas multas hay que aparcar la bicicleta en alguna de las estaciones antes de los t minutos y volver a pasar la tarjeta para volver a sacarla. De esta manera el contador de tiempo vuelve a 0.

Acaban de publicar dónde se encontrarán los puntos del bicicrucis de este año. Conociendo dónde se encuentran las estaciones de la Girocleta y el tiempo que hay entre ellas queremos saber si podemos hacer todo el recorrido sin que nos multen.

La ciudad de Girona consta de n puntos importantes. Cada uno de estos puntos puede contener una estación de la Girocleta. Estos puntos están conectados por m recorridos y cada recorrido conecta a dos puntos distintos. Además para cada recorrido r_i, conocemos el tiempo d_i que tardamos en ir en bicicleta de un lado a otro. Los recorridos se pueden atravesar en ambas direcciones. La salida y llegada del bicicrucis se hace desde el punto 0, donde sabemos que hay una parada de la Girocleta. Este año se van a visitar p de los puntos importantes. Ten en cuenta que no hace falta visitarlos en ningún orden concreto.

Se garantiza que todos los puntos están conectados por algún camino.

Entrada

La entrada empieza con un entero c indicando el número de casos.

Cada caso consiste en:

- Una línea con cuatro enteros n, m, p y t, indicando el número de puntos importantes de Girona, el número de recorridos, el número de paradas del bicicrucis y los minutos que se puede usar la Girocleta sin multa.
- La siguiente línea contiene n enteros e_i, donde $e_i = 1$ si en el punto $i-$ésimo hay estación o $e_i = 0$ en caso contrario.
- A continuación una línea con p enteros p_i indicando las paradas del bicicrucis.
- Las últimas m líneas contienen tres enteros a_i, b_i y d_i cada una indicando que hay un recorrido entre a_i y b_i que se puede recorrer en d_i minutos.

Restricciones

- $1 \leq c \leq 10$
- $1 \leq n \leq m \leq 10^4$
- $1 \leq p \leq n$
- $0 \leq a_i, b_i, p_i < n$
- $1 \leq d_i, t \leq 10^8$
- 11 puntos: todos los $e_i = 1$.
- 20 puntos: $m \leq 100$ y $p \leq 10$
- 22 puntos: $1 \leq m \leq 1000$
- 47 puntos: sin restricciones extra

Salida

Una línea con "Si" o "No" en función de si se puede hacer todo el bicicrucis usando la Girocleta sin pagar multas.

Ejemplos

Entrada	Salida
1 5 8 3 10 1 0 0 1 0 4 2 1 3 0 10 1 2 10 3 1 3 3 2 1 2 3 10 3 4 5 1 2 2 2 3 1	Si
1 5 6 3 9 1 0 0 0 1 1 0 3 3 0 7 1 2 8 3 1 8 3 2 9 2 3 6 3 4 8	No