Bounded Spanning Tree

É-te dado um grafo não dirigido e com ligações pesadas com n vértices e m ligações. Não existem lacetes no grafo (isto é, não existem ligações de um vértice para si próprio), mas podem existir múltiplas ligações entre um par de vértices.

O teu amigo disse-te o seguinte acerca deste grafo:

- Os pesos das ligações são números inteiros **distintos** do intervalo $[1, m]$. Por outras palavras, eles formam uma permutação dos inteiros de 1 até m.
- O peso da i-ésima ligação está no intervalo $[l_i, r_i]$ para cada i de 1 até m.
- As ligações com indíces $1, 2, \ldots, n-1$ (as primeiras $n-1$ ligações do input) formam uma árvore de suporte de custo **mínimo** (*minimum spannig tree*) do grafo.

Queres saber se isto é possível. Determina se existe um conjunto de atribuições de pesos a ligações tal que estas condições sejam obedecidas, e caso tal aconteça, descobre um desses conjuntos de atribuições.

Para relembrar, uma árvore de suporte de um grafo é um qualquer subconjunto das suas ligações que forme uma árvore (um grafo conexo com n vértices e $n-1$ ligações). A árvore de suporte de custo mínimo de um grafo é uma qualquer árvore de suporte com a menor soma possível de pesos das ligações de entre todas as possíveis árvores de suporte do grafo.

Input

A primeira linha contém um único inteiro t ($1 \le t \le 10^5$) - o número de casos de teste. Cada um dos casos de teste está descrito da seguinte maneira.

A primeira linha de cada caso de teste contém dois inteiros n e m ($1 \le n-1 \le m \le 5 \cdot 10^5$) - o número de vértices e de ligações, respetivamente.

A i-ésima das m linhas contém quatro inteiros u_i, v_i, l_i, r_i ($1 \le u_i < v_i \le n$, $1 \le l_i \le r_i \le m$) - indicando que existe uma ligação entre os vértices u_i, v_i e que o seu peso deve estar no intervalo $[l_i, r_i]$.

É garantido que para cada caso de teste, as ligações com índices $1, 2, \ldots, n-1$ formam uma árvore de suporte do grafo dado.

É garantido que a soma de m entre todos os casos de teste não excede $5 \cdot 10^5$.

Output

Para cada caso de teste, se não existe um array de pesos de ligações que satisfaça as condições, deves escrever '`NO`'' na primeira linha.

Caso contrário, na primeira linha deves escrever ''`YES`''. Na segunda linha deves escrever m inteiros $w_1, w_2, \ldots, w_m$ ($1 \le w_i \le m$, todos os w_i são **distintos**) - os pesos das ligações (onde w_i é o peso atribuído à i-ésima ligação do input).

Se existirem múltiplas respostas possíveis, podes escrever qualquer uma delas.

Podes escrever cada letra como maiúscula ou minúscula (por exemplo, ''`YES`'', ''`Yes`'', ''`yes`'', ''`yEs`'', ''`yEs`'' serão reconhecidos como respostas positivas).

Exemplo

Input:

```
3
4 6
1 2 1 3
1 3 2 6
3 4 1 2
1 4 2 5
2 3 2 4
2 4 4 6
4 4
1 2 2 2
2 3 3 3
3 4 4 4
1 4 1 4
5 6
1 2 1 1
2 3 1 2
3 4 2 4
4 5 6 6
1 4 4 6
1 4 5 6
```

Output:

```
YES
2 3 1 5 4 6
NO
YES
1 2 3 6 4 5
```

Pontuação

1. (4 pontos): $l_i = r_i$ ($1 \leq i \leq m$)
2. (6 pontos): A soma de m entre todos os casos de teste não excede 10
3. (10 pontos): A soma de m entre todos os casos de teste não excede 20
4. (10 pontos): $m = n - 1$, A soma de m entre todos os casos de teste não excede 500
5. (7 pontos): $m = n - 1$
6. (20 pontos): $m = n$
7. (11 pontos): A soma de m entre todos os casos de teste não excede 5000
8. (8 pontos): $u_i = i, v_i = i + 1$ ($1 \leq i \leq n - 1$)
9. (12 pontos): A soma de m entre todos os casos de teste não excede 10^5
10. (12 pontos): Nenhuma restrição adicional