

Adott egy súlyozott N csúcsú, M élű gráf: $N \leq 100\,000$, $M \leq 100\,000$. Adjuk meg minden e élre, hogy létezik-e e -t tartalmazó minimális feszítőfa. Minimális feszítőfának egy minimális összsúlyú feszítőfát nevezünk. Feszítőfa egy olyan fa, mely a gráf összes csúcsát, illetve az éleinek egy részhalmazát tartalmazza. Fának pedig egy olyan gráfot nevezünk, mely körmentes. Egy minimális feszítőfa tartalmazza e -t, ha egy olyan minimális feszítőfa, melynek az e az egyik éle. A program olvassa be a standard input első sorából N -et és M -et, majd a következő M sorból a k_i , v_i , s_i szóközzel elválasztott egészeket: azaz a k_i kezdőpontú, v_i végpontú, s_i súlyú éleket, és írja a standard output i -edik sorába az „Igen” szót, ha létezik olyan minimális feszítőfa, mely az i -edik élet tartalmazza, és a „Nem” szót, ha nem létezik ilyen.

Példa bemenet:	Példa kimenet:
6 10	Igen
1 2 2	Igen
1 6 1	Igen
1 5 3	Nem
4 1 5	Igen
2 6 2	Nem
2 3 5	Igen
4 3 4	Igen
3 5 4	Igen
4 5 4	Igen
5 6 3	

Pontozás és korlátok: A programhoz mellékelte, a helyes megoldás elvét tömören, de érthetően leíró dokumentáció 1 pontot ér. A programra akkor kapható meg a további 9 pont, ha bármilyen hibátlan bemenetet képes megoldani az 1 mp futásidőkorláton belül. Kapható részpontszám a 9 pontból, ha a program csak kisebb tesztesetekre tud lefutni időben. Az alábbi korlátok érvényesek az egyes részmegoldásokra:

- 2 pontért: $0 < M \leq 300$;
- további 3 pontért: $300 < M \leq 2\,000$;
- további 4 pontért: $2\,000 < M \leq 100\,000$.

Beküldendő egy tömörített `s79.zip` állományban a program forráskódja (`s79.pas`, `s79.cpp`, ...) az `.exe` és más, a fordító által generált állományok nélkül, valamint a program rövid dokumentációja (`s79.txt`, `s79.pdf`, ...), amely a fentiekén túl megadja, hogy a forrás mely fejlesztői környezetben fordítható.