

Submit: `voting.c` / `voting.cpp` / `voting.pas`
 Vstup: `stdin`
 Výstup: `stdout`

Časový limit: ?? s
 Paměťový limit: ?? MB
 Bodů: 100

Ve Spojeném království Velkého Bajtladnu se mají konat volby. Aby mohl občan království odvolit, musí jet do nejbližšího města, ve kterém se nachází volební místnost. Každý sice může (a měl by) jít k volbám, ale většina z nich je tak líná, že volit nepůjdou, pokud je nejbližší volební místnost dál, než jeden den cesty. Protože většina obyvatel bydlí ve vesnicích, které jsou poblíž silnice vedoucí mezi dvěma městy, bylo rozhodnuto, že volební místnost musí být vždy alespoň v jednom z měst, která jsou spojena silnicí. Moudrý král by chtěl mít ale volebních místností co nejméně, ale jeho rádci si s tím vůbec neví rady.

Úloha

Nalezněte pro zadané vstupní soubory, které dostanete k dispozici, co nejlepší rozmístění volebních místností.

Vstup

Dostanete seznam 10 vstupních souborů `voting1.in` až `voting10.in`. První řádka každého souboru obsahuje dvě čísla n a m ($1 \leq n \leq 50\,000$, $1 \leq m \leq 500\,000$), která určují počet měst a počet ulic v království. Města jsou očíslovaná čísla 1 až n . Každý z následujících m řádků obsahuje dvě čísla a_i a b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), která říkají, že je možné dostat cestovat mezi městy a_i a b_i za jeden den.

Výstup

Měli byste odevzdat jeden zazipovaný soubor, který obsahuje výstupní soubory pro zadané vstupy. Pro vstupní soubor `votingX.in` by se měl odpovídající výstupní soubor jmenovat `votingX.out`.

Výstup by měl obsahovat právě dva řádky. První musí obsahovat číslo k , které určuje počet volebních místností ve vašem řešení. Na druhém řádku by měl být seznam k měst, ve kterých se tyto volební místnosti mají nacházet. Počet získaných bodů závisí na velikosti vašeho řešení. Aby ale získalo vůbec nějaké body, musí být korektní. Pokud použijete y volebních místností, zatímco nejlepší známé řešení (které ale nemusí být stále nejlepší) používá b volebních místností, získáte $10(8b - 7y/b)^2$ bodů.

vstup	výstup
4 4	2
1 2	1 3
2 3	
3 1	
1 4	

