Zadanie: HER
Heros [A]

POTYCZKI ALGORYTMICZNE

Potyczki Algorytmiczne 2018, runda druga. Limity: 256 MB, 2-3 s. *11.12.2018 – 12.12.2018*

Najnowsza gra wyprodukowana przez Twoją firmę miała właśnie swoją premierę. Kolejne przygody przemierzającego Galaktykę herosa – wędrownego pogromcy kwazarów – sprzedają się nieźle i zbierają zupełnie dobre recenzje. Niektórzy tylko narzekają na fakt, że mechanika gry jest zbyt skomplikowana. Dobrze, że nic nie wiedzą o tym, jak naprawdę wyglądał proces produkcji: nieprzespane noce, niezdrowe odżywianie, napięta atmosfera i generator liczb losowych, którym trzeba było zastąpić jednego z kluczowych pracowników... zdecydowanie lepiej będzie, jeśli to wszystko nigdy nie wyjdzie na jaw.

W trakcie gry postać gracza może nauczyć się jednej z n umiejętności (takich jak *pilotowanie statków kosmicznych*, *magia żywiołów* lub *programowanie w PHP*). Umiejętności połączone są między sobą siecią zależności: danych jest m par (x, y) takich, że umiejętności y można się nauczyć dopiero po tym, kiedy pozna się umiejętność x. W sieci zależności nie ma cykli, czyli nie występuje sytuacja, że do nauczenia się umiejętności trzeba by było znać wcześniej ją samą. Mówiąc formalnie, umiejętności tworzą acykliczny graf skierowany (DAG).

Aha, czy wspominaliśmy już o generatorze liczb losowych, który musiał zastąpić odchodzącego pracownika? To właśnie on stworzył sieć umiejętności. Najpierw przyporządkował im w przypadkowy sposób numery $1, 2, \ldots, n$, a potem wylosował m różnych par (x, y) takich, że $x < y$ i z tych par utworzył zależności między umiejętnościami. To ucięło niekończące się dyskusje między programistami i znacznie przyśpieszyło produkcję, a także zmusiło Cię do ogłoszenia generatora Pracownikiem Miesiąca.

Stopień skomplikowania takiej sieci umiejętności to długość najdłuższego łańcucha, który można w niej odnaleźć – innymi słowy, najdłuższego możliwego ciągu takiego, że każda umiejętność w nim wymaga poprzedniej. Zbyt duży stopień skomplikowania powoduje, że graczom ciężko się odnaleźć w mechanice gry, a dodatkowo schemat sieci źle wygląda na ekranie. Ponieważ nikt (poza generatorem liczb losowych, który jednak nie jest skłonny do zeznań) nie rozumie obecnej struktury sieci, zapadła radykalna decyzja: usuń k spośród umiejętności (wraz z ich wszystkimi zależnościami) tak, aby stopień skomplikowania pozostałej sieci zmniejszył się do możliwie najmniejszej wartości.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby całkowite n, m oraz k ($1 \leq n \leq 300$, $0 \leq m \leq 400$, $0 \leq k \leq min(n, 4)$), oznaczające odpowiednio aktualną liczbę umiejętności, liczbę zależności oraz liczbę umiejętności do usunięcia. Umiejętności numerujemy od 1 do n.

W kolejnych m wierszach znajdują się po dwie liczby x i y ($1 \leq x < y \leq n$), oznaczające że przed nauczeniem się umiejętności y należy poznać x. Możesz założyć, że zależności zostały wygenerowane w sposób losowy, metodą opisaną w treści zadania.

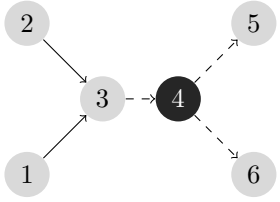
Wyjście

Na wyjście należy wypisać jedną liczbę – najmniejszy możliwy stopień skomplikowania sieci umiejętności po usunięciu co najwyżej k spośród nich.

Przykłady

Dla danych wejściowych:

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



poprawnym wynikiem jest:

```
2
```

Wyjaśnienie do przykładu: Jeśli usuniesz umiejętność numer 4, to najdłuższy łańcuch będzie miał długość 2 (na przykład $1 \to 3$, albo $2 \to 3$). Tak więc najmniejszy możliwy stopień skomplikowania wynosi 2 – nie jesteś w stanie usunąć jednej umiejętności tak, aby otrzymać stopień 1.

Natomiast dla danych wejściowych:

```
3 3 3
1 2
1 3
2 3
```

poprawnym wynikiem jest:

```
0
```