

Модное платье

Город в полном составе готовится к времени фестивалей, кабаре и свадеб. Все женщины города с нетерпением ждут "Главный бал года", который проводится в последнюю субботу праздника. "Главный бал" это знаковое событие, к которому начинают готовиться загодя до начала. Один из главных вопросов который задает себе каждая девушка города: "Что мне одеть, что бы поразить всех своей красотой?". Эта проблема особенно ощутима в связи с тем, что после этого праздника следующий будет только спустя несколько месяцев. И в течение этого скучного времени дамам приятно вспоминать как обворожительно и оригинально они выглядели во время бала. И само собой Вам следует запомнить, что если две подружки появятся в одинаковых платьях, то они будут объектом шуток и приколов на ближайшие несколько лет.

Как и в большинстве сложных проблем между женщинами в них завязли и мужчины города. И каждый день каждая женщина города спрашивает у своего парня совета, что именно ей одеть. К сожалению мужчины совсем не подкованы в вопросах моды и поэтому они обратились за помощью к вам.

У вас также не очень много опыта в области моды, однако, как специалист по компьютерам и одаренный человек, который никогда не сдается, вы знаете, как решить любую проблему. Вы смоделировали ситуацию следующим образом:

- осталось $1 \leq k \leq 20$ дней до "Главного бала",
- платья бывают нескольких цветов, которые вы обозначили целыми числами, начиная с нуля,
- взаимоотношения между женщинами задаются в виде графа (смежные узлы обозначают двух подруг),
- каждая девушка хочет, что бы её платье отличалось от платьев её подруг, иначе она будет очень несчастна.

У вас есть идея, дать каждому мужчине набор простых правил, с помощью которых они смогут определить лучший, наиболее уникальный, цвет платьев для их пассий. У вас не так много времени, поэтому все мужчины будут действовать по одному и тому же алгоритму, придуманному вами. Из-за того что не каждый из мужчин использует компьютер, вам следует использовать очень простой язык для описания своего алгоритма.

Задание

Напишите алгоритм, который в случае удачного применения мужчинами, будет гарантировать, что несчастными после бала останутся как можно меньше женщин.

Краткое описание языка, который вам следует использовать приведено ниже:

- каждая программа это набор правил (rule),
- каждое `< rule >` представимо в виде:

```
< rule > ::= if < comp_condition > then { < comp_command > };  
  
< comp_command > ::= < command >;
```

`< comp_command > ::= < command >; < comp_command >`

`< comp_condition > ::= < condition >`

`< comp_condition > ::= not < comp_condition >`

`< comp_condition > ::= (< comp_condition > or < comp_condition >)`

`< comp_condition > ::= (< comp_condition > and < comp_condition >)`

Учтите, что после каждой `< command >` и `< rule >` вы должны поставить точку с запятой.

- `< command >` это просто присваивание вида:

`< command > ::= < symbol > := < expr >`

- `< symbol >` может быть одной из следующих строк:

- компонента состояния: `color`, `a`, `b`,

- дополнительные локальные переменные: `I0`, `I1`, `I2`, `I3`, `I4`, `I5`, `I6`, `I7`, `I8`, `I9`.

- Выражение `< expr >` следует составлять используя:

- записываемые символы (приведенные выше) и несколько специальных переменных (`read-only`), которые дают следующую информацию о графе:

`deg` степень текущей вершины (`degree of vertex`)

`delta` максимальную степень вершины в графе (`maximum vertex degree`)

`n` количество вершин (`number of vertices`)

`m` количество ребер (`number of edges`)

`nr` количество доступных цветов

`daysleft` количество дней до бала

- операторы (в порядке приоритета):

`+` `-` арифметические плюс и минус

`*` `/` `%` арифметические умножение, деление и остаток от деления

- функция `rnd(< expr >)`, возвращает случайное число между 0 и `< expr > - 1` включительно.

- функции `mina`, `minb`, `minc` возвращают минимальное значение переменной `a`, `b`, `color` соответственно, среди всех вершин смежных с данной (или минимальное целое число для изолированной вершины). Функции `maxa`, `maxb`, `maxc` действуют также, но для максимальных значений переменных.

- `< condition >` это логическое выражение заданное в одной из следующих форм:

`< condition > ::= < expr > < operator > < expr >`

`< condition > ::= < exist-operator > (< expr >)`

Бинарный оператор `< operator >` это один из операторов сравнения `==`, `<`, `or`. Унарный оператор `< exist-operator >` это одна из следующих функций: `Eaeq`, `Ebeq`, `Eseq`, которые возвращают `true`, если для некоторого из соседей данного ребра значение переменной `a`, `b`, `color`, соответственно, равно `< expr >`.

Алгоритм будет применяться каждым мужчиной города, каждый день. Более формально: в начале процесса все переменные будут иметь случайные значения. Каждое утро каждый мужчина присвоит 0 своим переменным `I0, I1, ..., I9`. Точно в полдень каждая женщина придет за советом. Мужчина сделает все, что сможет, что бы помочь свое девушке: он просмотрит все правила сверху вниз и будет повторять процесс до тех пор, пока хотя бы один из `IF` возвращает `true`. (Однако, если ему придется делать это больше чем сто раз то он сдастся). Затем он скажет своей леди цвет, который он выбрал для неё.

Каждый вечер мужчины встречаются в баре и когда им уже нечего обсуждать, они обмениваются информацией о том как их девушки им досаждают и какие значения a , b и $color$ они им посоветовали сегодня. (Заметим так же что именно эти значения используют функции E_eq , $min_$ и $max_$ на следующий день.)

Наконец, в день бала женщины одевают платья в соответствии с последним советом своих мужчин. И приходит время оценить эффективность предложенного вами алгоритма. (Это одна из тех задач, которая в случае плохого исполнения может привести к линчеванию программиста кровожадной толпой.)

Начисление очков

Всего на вход вашей программе будет подано 50 тестов. Используемые в тестах графы имеют не более 25 вершин. Общее количество очков, полученное за ваше решение, будет равно сумме очков за каждый отдельный тест. За каждый тест количество очков будет равно отношению количества правильно раскрашенных вершин (т.е. таких что $0 \leq color < nr$ и $Eseq(color)$ равно $false$) к общему числу вершин в графе. Если все вершины раскрашены правильно, то дополнительно вы получите бонус равный $1/(1 + \text{максимальный использованный цвет})$ очков.

Программа получит статус `Compilation Error`, если она не будет корректно интерпретирована из-за синтаксических ошибок. Если в некоторый день правила для заданной вершины не смогут быть обработаны в течении 100 итераций или весь процесс моделирования займет более 60 секунд, то ваша программа получит статус `Time Limit Exceeded`.

Пример

Решение следует отправлять на языке `TEXT`, [программа судья](#) обработает ваше решение. Ниже приведен пример простой программы состоящей из одного правила.

Код:

```
if (l0==0 and not Eseq ((color-1)%n)) then {color:=(color-1)%n; l0:=1;};
```

Очки:

5.491