

Mice (Решение)

Това е другата задача, определена като "трудна" в темата. За нейното решаване се изискваха отново що-годе стандартни алгоритъма, но малко по-сложна теория и известна креативност, при строенето на графа.

Първо, очевидно задачата беше графова. Второ, клонирането на мишките и тръгването по всеки път беше много слаб опит за дегизиране на това, че се пита колко пътища има с дължина до K от началния до крайния връх. А алгоритъм за намиране на този брой има.

Добре де, поне има за намиране на броя пътища с дължина точно K . Алгоритъмът е следния: взимаме графа, представен като матрица на инцидентност, чиято стойност в клетка (i, j) е 0, ако нямаме ребро, и 1, ако имаме такова между върховете i и j . Тази матрица вдигаме на степен K , за да намерим колко пътища има с дължина точно K между произволни два върха. След вдигането на степен, клетка (i, j) дава именно броя пътища с дължина точно K от връх i до връх j .

Тъй като K е сравнително голямо, трябва да имплементираме бързо вдигане на степен на матрици. Като цяло това е алгоритъмът за бързо вдигане на степен на число, използващ алгоритъма за умножение на матрици, вместо обикновено умножение. Нищо кой-знае колко сложно.

Модификацията за "до K " също не е сложна - просто трябва да сложим ребро (примка) от крайния връх до себе си, за да симулираме "събирането" на мишки в него.

Обаче. Освен клонирането на мишките в самия граф, Ели пуска мишка на всяка секунда. Това пък можем да симулираме, като направим примка от началния връх до самия себе си.

Обаче. Ели не пуска мишки във времена, кратни на 4. Това можем да решим, като от отговора, намерен с горното решение, извадим отговора, който бихме получили ако Ели добавяше мишка на всеки 4 секунди. Това се симулира, като от началния връх добавим път до самия себе си, с дължина, равна на 4.

Обаче. Ели не пуска мишки и във времена, кратни на 7. Това решаваме точно като горния проблем, само че пътят от началния връх до себе си ще е с дължина 7.

Обаче. Сега пък Ели не пуска по две мишки във времена, кратни на 28 ($4 * 7$). За да се справим с това пък ще прибавим броя мишки, които биха излезли ако се появяваха по веднъж на всеки 28 секунди.

И вече сме готови. Реално правим включване/изключване със само две

числа - 4 и 7. Така де, и 1. Прибавяме 1, вадим 4, вадим 7, и прибавяме 28. Това е така, понеже 1 има само един множител (самото 1), 4 има два множителя (1 и 4), 7 има два множителя (1, 7) и 28 има три множителя (1, 4, 7). Както знаем, при метода на включването и изключването, прибавяме нечетните и вадим четните.

Трик: Вместо да пускаме вдигане на матрица на степен четири пъти - по веднъж за всяка от съпките на включването и изключването - можем да направим малко по-голям граф, чиято матрица да вдигнем на степен само веднъж. Така, макар и правейки матрицата малко по-голяма, реално алгоритъмът върви около два пъти по-бързо (0.75s срещу 1.53s при time limit 2.0s).