

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА WORDROW

Това е една по-нестандартна задача, която е относително лесна за задаване (постановката е много проста), но не тривиална за решаване. И все пак е далеч от това колко трудна първоначално си представях, че ще бъде, така че можете да се радвате! :)

Може би най-простото решение тук би било да пазим битова маска кои от буквите вече са задраскани (като броят задраскани ни казва и кой от двамата играчи е на ход). Ползвайки проста теория на игрите ("ако имаме ход, с който вкарваме противника в губеща позиция, значи текущата ни позиция е печеливша") можем да разделим различните позиции (битови маски) на "печеливши" и "губещи" за Ели. Съответно, накрая можем да проверим дали позицията с битова маска 0 (тоест никоя буква не е задраскана) е печеливша или губеща за първия играч. Ако е печеливша, значи Ели печели. В противен случай Крис печели.

Това решение обаче изисква твърде много памет. Може би максимума, който може понесе, са дъски с около 25 клетки (примерно 5 на 5 или 4 на 6). Все пак, по-добрите състезатели биха почнали с това решение – първо за да имат някакви гарантирани точки, и второ, тъй като това решение вървши чудесна работа да тествате по-сложни такива (в интерес на истината и аз подходах така докато подготвях задачата). Това решение е със сложност $O(T * 2^{N*M})$ и би хванало около 30 точки (даже повече, с подходящи оптимизации).

Как, тогава, да се справим с дъски, които са до 10 на 10? За да подходим към по-големите дъски трябва да направим едно много важно наблюдение (не особено трудно, но все пак) – тъй като финалната дума трябва да е палиндром, то можем да разделим колоните на матрицата по "двойки" (освен средната, ако броят колони е нечетен, която е самичка). Така получаваме $M / 2$ на брой подзадачи, всяка съдържаща $2 * N$ клетки. Забележете, че тъй като $2 * N$ е около 20, то отново можем да ползваме битова маска и всъщност да ползваме горното решение – само че за по-малки подзадачи. Така за всяка двойка симетрични колони, ще знаем дали Ели би спечелила или Крис би спечелила.

Тъй като обаче имаме няколко подзадачи, които не са напълно независими (не е гарантирано, че двете момичета няма да "сменят" двойките колони, в които зачеркват, по време на играта), трябва да измислим и стратегия кога Ели побеждава. Това става като усложним съвсем малко решението с битовите маски – не само ще казваме дали за дадена двойка колони Ели побеждава като първа (демек успява да докара две еднакви букви във всяка от двете симетрични колони независимо как играе Крис), ами ще можем да кажем дали Ели би победила като първа **ИЛИ** като втора.

Тъй като тя трябва да започне играта, то в крайна сметка трябва да има поне една двойка симетрични колони, в които тя да може да спечели като първа. Алтернативно, ако M е нечетно, тя би могла да победи в средната колона. Забележете, че там Ели винаги побеждава, както като първа, така и като втора, тъй като накрая остава само една буква, която не прецаква палиндрома. Всички останали двойки колони тя трябва да може да победи както като първа, така и като втора, тъй като ако Ели сложи в някоя двойка колони, Крис може да я "прецака", като веднага сложи в друга двойка колони и накара Ели да играе там като втора.

Частен случай тук е когато M е нечетно (има средна колона), а N е четно (тоест трябва да се зачеркнат нечетен брой символи в средната колона). Тогава Ели може да започне задраскване в средната колона и да продължи да задрасква там, ако Крис също задрасква там, или пък да прави симетрични на Крис зачерквания, когато Крис задрасква в някоя от другите колони. Тъй като в двойка симетрични колони винаги има четен брой символи за зачеркване, Ели може да си остане втора във всяка "под-игра". В този частен случай е достатъчно Ели да може да спечели всяка двойка колони като втора (но пада ограничението да може да ги спечели и като първа).

Това решение е със сложност $O(T * M * 2^{2N})$ и работи достатъчно бързо за 100 точки.

Интересен факт е, че може да се изведе и решение, което работи за $O(T * N * M)$, което би работело за много по-големи дъски (то се базира на броене на еднакви букви в симетрични колони). За съжаление, то разчита на разглеждане на много частни случаи, което обикновено са много досадни задачи и реших да не го изисквам от състезателите.

Автор: Александър Георгиев