
Решение на Антония Няголова (annieharris)

В това решение функцията `greedy(int from, int when)` се състои от един цикъл `for(int t=from; t<N*N; t++)`. В този цикъл първо се премахват всички царици сложени преди и се слага царица на полето с индекс `t`. След това се поставят царици навсякъде, където може, докато няма колизии. Накрая е частта, в която се поставят царици на най-изгодните места като се гледа стойността на $(score / coll)$ да е максимална, където `score` е точките, които полето носи, а `coll` - броят на колизиите, които се получават, ако там се сложи царица. Тогава е позволено да има колизии, стига да не надвишават `K`. Функцията спира, когато общото време през което програмата е била пусната, надвиши стойността на променливата `when` (в милисекунди). `When`, разбира се, е броят милисекунди след изпълнението на програмата, когато функцията трябва да прекрати своето действие, за да се избегне `time limit`. Тази практика е шикоро разпространена в маратоните и е позната под неофициалното наименование "time-бомба."

На всяка стъпка на главния цикъл, както и когато излиза, функцията проверява дали текущото решение е по-добро от най-доброто намерено, и ако е, го запазва.

Първоначално, полетата са подредени така, че полето на позиция (i, j) , т.е. ред `i` и колона `j`, е с индекс $i*N + j$. Функцията `greedy` се пуска с параметри `from=0` и `when=2250`. Така тя използва почти половината от времето, през което програмата ще работи максимално, защото ограничението за време е 5 секунди (5000 милисекунди).

След това полетата се сортират по точките, които носят (тези точки зависят единствено от стойностите в матрицата, които никога не се променят). Тогава `greedy` се пуска с параметри `from=N` и `when=4900`. Чрез опити е установено, че след сортирането, когато `from` е равно на `N`, а не на `0` (с други думи се започне от полето, което е $(N+1)$ -во по брой точки в низходящ ред), се достигат по-добри резултати.