

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА RUNAWAY

Всеки състезател много лесно може да се досети за наивното решение на задачата. При всеки опит за бягство се претърсва цялата матрица. След намиране на някой от изходите пресмятането на търсеното разстояние е тривиално. Лесно се вижда, че това решение има голяма сложност и няма да покрие всички тестови примери.

Можем да се възползваме от това, че по условие числата в матрицата ни е сортирана. Тогава лесно можем да се десетим, че при всеки опит за бягство търсеният изход можем да намерим с двоично търсене. Реализацията на това решение има сложност от порядъка на $O(k \cdot n \cdot \log n)$, но това отново не е достатъчно бързо.

Нека разгледаме числото x – номера на стаята, от която може да се избяга и числото p – числото, което се намира в най-горната дясна клетка на матрицата – $[1][n]$. В следствие от сортировката на матрицата имаме следните три ситуации:

1 – Числото x е равно на числото p . Тогава пресмятаме пътя от p до x и отпечатваме резултата.

2 – Числото x е по-голямо от числото p . Тогава числото x няма как да се намира в същия ред. Следователно можем да „скъсим“ матрицата, като не разглеждаме най-горния ред.

3 – Числото x е по-малко от числото p . Тогава числото x няма как да се намира в същата колона. Следователно можем да „скъсим“ матрицата, като не разглеждаме най-дясната колона.

Така на всяка стъпка, в която не намерим числото, смаляваме матрицата. Максималният брой стъпки, за които можем да намерим число, е $2 \cdot n$ и сложността на алгоритъма става от порядъка на $O(k \cdot 2 \cdot n)$;

Анализ: Петър Петров