**АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА**

**LOTARY**

Да забележим следното:

Ако имаме вероятностите да се намираме във всеки от секторите след M завъртания (ще означавам с a\_m,i), можем да получим вероятностите a\_m+1,i за време N^2 по следния начин: a\_m+1,i = a\_m,i\*p\_0 + a\_m,i-1\*p\_1 + .... + a\_m,i+1\*p\_n-1.

Наратко вероятността да съм дошъл със завъртане на 0 (от i) + вероятността да съм дошъл със завъртане на 1 (от i-1) + .... вероятността да съм дошъл със завъртане на n-1 (oт i+1).

Това ни дава решение за време O(N\*N\*K).

Значи очевидно можем да получим a\_m+1 от a\_m за N^2 време, обаче ако бъдем хитри може да се сетим че всъщност можем да получим a\_2m от а\_m.

Как? Ами важното наблюдение е че a\_m,i не просто ни дава вероятността след m завъртания да сме в i-тия сектор, ами ни дава вероятността с m завъртания да завъртим с още i-1 сектора.

От тук насетне ще броя секторите от 0 за удобство. Т.е. a\_m,i ще ни дава вероятността да завъртим с още i сектора (а не i-1).

Т.е. по подобен начин можем да изразим a\_2m,i = a\_m,i\*a\_m,0 + a\_m,i-1\*a\_m,1 + .... + a\_m,i+1\*a\_m,n-1.

Това вече ни дава начин да изчислим отговора за време O(N\*N\*logK) със стратегия подобна на тази при бързо повдигане на степен.

Важно да се отбележи и че ако разглеждаме матрицата на вероятностите на преходите от състояние в състояние можем просто да повдигнем тази матрица на степен K и отново ще получим отговора. Това дава решение O(N^3\*logK). Причината да ни се получава за време О(N^2\*logK) е че матрицата е циркулантна а циркулантните матрици могат да се умножават по-бързо.

И сега идва въпроса как да избягаме от това N^2. Трябва по бързо от a\_x и a\_y да можем да получим a\_x+y. Сега да разгледаме формулата която бяхме извели а именно:

а\_x+y,i = a\_x,0\*a\_y,i + a\_x,1\*a\_y,i-1 + ... + a\_x,i\*a\_y,0 + a\_x,i+1\*a\_y,n-1 + ... + a\_x,n-1\*a\_y,i+1

//За човек с повече знание на теория още тук би могло да стане ясно че dft(a\_x+y) = dft(a\_x)\*dft(a\_y), където dft е дисктретната трансформация на Фурие и \* е поелементно умножение. Това веднага ни дава време за изчисление O(N\*logN) и съответно решение на цялата задача за O(N\*logN\*logK), но нека се придържаме към по-просто разбиране.

Да разделим този сбор на две части:

L\_i = a\_x,0\*a\_y,i + a\_x,1\*a\_y,i-1 + ... + a\_x,i\*a\_y,0 и

R\_i = a\_x,i+1\*a\_y,n-1 + ... + a\_x,n-1\*a\_y,i+1

Тези два сбора от произведения доста видно се получават при умножение на полиноми (ще разглеждам полиномите като вектор от коефициентите си).

В частност при умножението на полиномите a\_x и a\_y, L\_i е коефициента пред i-тата степен и R\_i е коефициента пред n+i-тата степен.

Т.е. ако получим произведението на полиномите R. То a\_x+y,i = R\_i + R\_i+n, А R може да получим за време O(N\*logN) използвайки стандатрния метод за бързо умножение на полиноми с FFT.

Това вече ни дава решение O(N\*logN\*logK).

*Автор: Иво Дилов*