Конфигуреум генератор на ролки за слот машина с рестрикции

1. Въведение

Хаотичното разбъркване (без детерминизъм) на колекция от елементи е не тривиална задача. Ако считаме ендаквите елементи от колекцията за неразличими, то сортирането ще е единствено, докато разбърквания ще са точно $\frac{n!}{k_1! \times k_2! \times \ldots \times k_j!}$ на брой, където n е броя на всички елементи в колекцията, а $k_1, \ldots, k_j$ са съответно броевете на повторенията на различимите j елементи в колекцията.

Например, броят на разбърквания на редицата (колекцията) от числови елементи

$$3, 1, 2, 1, 1, 2$$

е равен на $\frac{6!}{3! \times 2! \times 1!} = 60$, докато сортирането е само едно: $1, 1, 1, 2, 2, 3$ (1-ците, 2-ките и 3-ките са неразличими елементи).

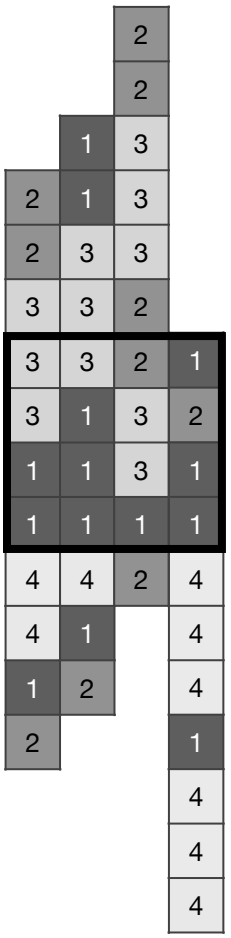
Въвеждането на рестрикции върху дадено разбъркване може както да улесни, така и да усложни алгоритъма за осъществяването му, но по отношение на сложност по време за работа – винаги ще нараства при всяка рестрикция в средния случай.

2. Специфична механика, в която са необходими рестрикции

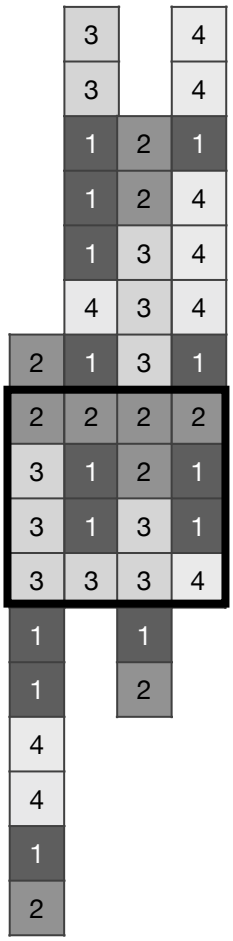
Да разгледаме слот игра с клъстерна стратегия за изплащане. В този тип стратегия играта изплаща спрямо таблица на изплащанията, тогава и само тогава, когато не по-малко от k на брой неразличими символи имат поне една съседна стена с поне един друг символ от клъстера (от гледна точка на математиката – символите са просто квадрати с идентификационен номер). От тук идва и името на стратегията за изплащане.

Нека номерираме символите с числата от 1 до n, където n е броя на всички различни символи по ролките. Да разгледаме например печелившо множество от ролки, за което $n = 4, k = 5$ и екрана е с размерност 4×4 (под печелившо множество от ролки ще разбираме такова множество, което може да генерира поне една печалба от всички видове разположения на ролките):

2	1	2	1
2	1	2	1
3	3	3	4
3	3	3	4
3	3	3	4
1	1	2	1
1	1	2	4
4	1	3	4
4	4	3	4
1	1	1	1
2	2	2	2



печеливши единици



печеливши двойки и тройки

В първия случай с печелившите единици, от четвъртата ролка са взети последните два символа и са се допълнили с първите два от ролката. Аналогично и за втората и четвъртата ролки от втория случай.

Искаме да предоставим алгоритъм, които при вход – множество от множества от бройки за всеки символ по дадена ролка създава непечелившо множество от ролки за клъстерна слот машина. Алгоритъмът трябва да приема като допълнителни параметри:

- n – брой на различимите елементи (tiles) по ролките;
- k – минимален брой елементи в печеливш клъстер;
- sl_i, sr_i – минимален и максимален брой за стакнати символи на i-тата ролка;
- re – максимален брой опити за избор на символ, който да се постави в ролката, преди да се траверсира цялата ролка, за да се намери подходящо място за текущо избрания символ (retries);
- d – минимална дистанция между веки два неразличими елемента по ролката, които са от различни стакове;
- m – минимален брой съседни ролки, в който да липсва поне един символ от всеки вид в поне една от тях. (Тази рестрикция служи за улеснение на алгоритъма и може да се валидира програмно. Чрез нея може да се задава грубата форма на клъстера – дали да е по-вертикална или по-хоризонтална (разпрозтираща се на повече ролки))

Примерен вход	Примерен очакван изход
[0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0], [0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0], [0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0]	[2, 2, 2, 9, 9, 9, 3, 3, 3, 5, 5, 5, 8, 8, 8, 5, 5, 5, 2, 2, 2, 3, 3, 3, 5, 5, 8, 2, 2, 2, 6, 6, 6, 2, 2, 3, 3, 3, 6, 6, 6, 3] [1, 6, 9, 1, 4, 7, 1, 3, 6, 4, 9, 6, 3, 7, 9, 3, 1, 4, 3, 1, 6, 3, 1, 4, 6, 1, 4, 3, 6, 7, 4, 1, 3, 4, 7, 3, 1, 7, 4, 1, 3, 4, 1, 3, 4] [2, 2, 2, 4, 4, 4, 2, 2, 7, 7, 7, 4, 1, 1, 1, 5, 5, 5, 8, 8, 8, 4, 4, 4, 8, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 2, 5, 5, 5, 1, 1, 7, 7, 5, 5, 4, 4, 4] [9, 3, 2, 5, 9, 2, 8, 6, 2, 3, 9, 2, 8, 3, 6, 2, 8, 5, 2, 3, 8, 6, 3, 5, 6, 3, 5, 6, 3, 2, 6, 5, 2, 3, 5, 2, 3, 5, 2, 3, 5, 2] [6, 6, 6, 4, 7, 7, 7, 9, 9, 9, 1, 1, 1, 3, 3, 3, 4, 4, 4, 1, 1, 1, 7, 7, 3, 3, 3, 1, 1, 1, 6, 6, 6, 3, 3, 3, 4, 4, 4, 1, 1, 3, 4, 4, 4] [7, 8, 1, 7, 2, 1, 4, 7, 5, 2, 4, 1, 2, 7, 4, 5, 7, 1, 8, 4, 2, 1, 4, 2, 5, 8, 1, 2, 4, 8, 2, 4, 5, 1, 2, 4, 5, 1, 2, 5, 1, 4, 2, 5, 4, 1, 2, 5, 1] [2, 2, 2, 5, 5, 5, 2, 2, 6, 6, 6, 5, 5, 3, 3, 3, 6, 6, 6, 2, 2, 2, 3, 3, 3, 8, 8, 8, 3, 3, 3, 2, 2, 2, 9, 9, 9, 3, 8, 5, 5, 5] [9, 6, 1, 4, 9, 3, 1, 4, 9, 1, 3, 4, 1, 7, 3, 6, 7, 3, 6, 7, 3, 4, 6, 1, 7, 4, 6, 1, 3, 6, 7, 1, 4, 3, 1, 4, 3, 1, 4, 3, 1, 4, 3, 1, 4] [8, 8, 8, 1, 1, 1, 4, 4, 4, 7, 7, 7, 8, 2, 2, 2, 7, 7, 4, 4, 4, 5, 5, 5, 1, 1, 1, 2, 2, 2, 1, 1, 1, 5, 5, 5, 4, 4, 4, 5, 5, 2, 2, 2, 4, 1, 1, 2, 2]

В примера по-горе ролките с четни номера (броене с база 0) са стакнати с $[3, 3]$, а ролките с нечетни номера са стакнати с $[k-3-1, k-3-1]$, т.е. с 1, ако приемем, че $k = 5$ и дистанцията между всеки два неразличими tile-а да е поне $2 = d - 1$.
Аналогично може да зададем ролките на четни номера да са стакнати с $[m_1, m_2]$, а ролките с нечетни номера да са стакнати с $[k - m_2 - 1, k - m_2 - 1]$ и отново да получим друг вид непечеливши стакнати ролки и дистанцията между всеки два стака с нерачзличими tile-а да е поне m_1.

3. Алгоритъм

1. Намапваме id-то на всеки tile със съответния му брой в сортиран дървовиден map (работи по-бързо за малко на брой ключове отколкото хеш мап-а) за всяка ролка.
2. Създаваме двусвързан списък, в който ще генерираме резултата за съответната ролка, тъй като ще правим операции по вмъкване на елемент в края, в началото и на произволен индекс.
3. Създаваме втори дървовиден map, в който ще съхраняваме за ключ id на tile, а за стойност – преди колко позиции назад се среща първия tile със същото id.
4. Докато има tile-ове за поставяне по ролките – избираме произволно число от 0 до броя на всички tile id-та, които са останали за поставяне минус 1. След това пропускаме това произволно число от стриима на ключовете на map-а (дървовидния map е сортирана структура) и присвояваме в променлива id-то на текущия за поставяне tile. Взимаме броя на този tile и ако той е равен на 0 – изтриваме ключа с това id от map-а и се връщаме в началото на 4. (в началото на while цикъла).
5. Избираме размера на текущия стак и задаваме броя на повторенията да е минималното от текущия стак и броя, с който разполагаме от tile-а с това id.
6. Ако последно поставения tile от този вид е не по далеч от зададената минимална дистанция се връщаме отново на стъпка 4. Преди да се върнем на стъпка 4 обаче, проверяваме дали не сме направили максималния предефиниран брой такива връщания и ако да – значи няма как да изберем стак от tile-ове, които да поставим най-накрая в списъка с tile-ове и за това преминаваме към поставяне на стека от текущия tile на първата допустима позиция в началото на списъка. Ако не успеем да поставим стека от текущия tile и по този начин за повече от предефиниран максимален брой произволни tile-ове – чак токава връщаме празен списък, с който да индикираме че сортирането с така конфигурираните рестрикции се е провалило. Актуализираме броя на tile-овете и последно срещаните позиции на всеки tile, който се е появявал в ролката до сега.
7. В противен случай сме поставили стека с tile-ове и само актуализираме броя на tile-овете и последно срещаните позиции на всеки tile, който се е появявал в ролката до сега.
8. Проверяваме дали в началото и края на ролката няма конфликти (т.е. дали няма tile-ове, които са на дистанция по-малка от минималната позволена, като свържем началото с края на двусвързания списък).

Този алгоритъм ни позволява да генерираме непечеливши ролки с максимален стак за слот машина с клъстерна стратегия за изплащане, ако изплаща за минимум k символа в клъстера, където $k \geq 5$.

Нека например $k = 7$. Примерни непечеливши ролки със същите пропорции на броя символи както от предходния пример ще са следните:

Примерен вход	Примерен очакван изход
[0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0], [0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0], [0,11,10,0,8,6,0,4,3], [11,0,10,10,0,6,5,0,3], [11,11,0,10,8,0,5,4,0]	[8, 8, 8, 8, 2, 2, 2, 2, 6, 6, 6, 6, 2, 2, 2, 2, 6, 6, 9, 9, 9, 3, 3, 3, 3, 5, 5, 5, 5, 2, 2, 2, 3, 3, 3, 3, 5, 5, 5, 5, 3, 3] [6, 6, 7, 7, 1, 3, 3, 4, 4, 1, 1, 9, 9, 6, 6, 4, 4, 1, 1, 6, 6, 9, 4, 4, 3, 3, 7, 7, 1, 1, 3, 3, 7, 4, 4, 1, 1, 3, 3, 4, 4, 1, 1, 3, 3] [5, 5, 5, 5, 7, 7, 7, 7, 2, 2, 2, 2, 4, 4, 4, 4, 5, 5, 5, 5, 8, 8, 8, 8, 2, 2, 2, 2, 1, 1, 1, 1, 7, 4, 4, 4, 4, 1, 1, 1, 1, 2, 2, 2, 4, 4, 1, 1, 1] [2, 2, 8, 8, 3, 3, 2, 9, 9, 5, 5, 2, 2, 3, 3, 8, 8, 6, 6, 3, 3, 2, 2, 5, 5, 9, 6, 6, 3, 3, 2, 2, 6, 6, 5, 5, 2, 2, 3, 3, 5, 5] [9, 9, 9, 4, 4, 3, 3, 3, 3, 6, 6, 6, 6, 7, 7, 7, 7, 3, 3, 3, 3, 4, 4, 4, 4, 1, 1, 1, 1, 6, 6, 1, 1, 1, 1, 3, 3, 1, 1, 1, 4, 4, 4, 4, 7] [4, 4, 5, 5, 2, 2, 4, 4, 1, 1, 8, 8, 2, 4, 4, 1, 1, 2, 2, 8, 8, 1, 1, 2, 2, 5, 5, 1, 4, 4, 2, 2, 7, 7, 4, 4, 5, 5, 1, 1, 7, 7, 5, 5, 1, 1, 2, 2, 7] [6, 6, 6, 6, 8, 8, 8, 8, 6, 6, 9, 9, 9, 5, 5, 5, 5, 2, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 2, 3, 3, 3, 3, 5, 5, 5, 5, 3, 3, 2, 2, 2] [7, 7, 3, 3, 1, 6, 6, 9, 9, 4, 4, 1, 1, 7, 7, 9, 6, 6, 3, 3, 4, 4, 7, 1, 1, 6, 6, 4, 4, 3, 3, 1, 1, 4, 4, 3, 3, 1, 1, 4, 4, 3, 3, 1, 1] [7, 7, 7, 7, 4, 4, 8, 8, 8, 8, 2, 2, 2, 2, 7, 5, 5, 5, 5, 2, 2, 2, 2, 5, 5, 5, 5, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 4, 4, 4, 4, 1, 1, 1, 4, 4, 4, 4]

В този пример обаче, ролките с **четни** номера са стакнати с $[4, 4]$ (дистанцията между всеки два неразличими tile-а е поне 2), а ролките с **нечетни** номера са стакнати с $[k - 4 - 1, k - 4 - 1] = [2, 2]$ и дистанцията между всеки два неразличими tile-а е поне 4.

Аналогично може да зададем ролките на четни номера да са стакнати с $[m_1, m_2]$, а ролките с нечетни номера да са стакнати с $[k - m_2 - 1, k - m_2 - 1]$ и дистанцията между всеки два неразличими tile-а в четните ролки да е поне m_1, а в нечетните да е поне $k - m_1 - 1$ и отново да получим друг вид непечеливши стакнати ролки.