Баннерная крутилка с весами

Постановка задачи

Необходимо из n различных взвешенных баннеров выбрать k различных баннеров в соответствии с весами.

На вход:

набор взвешенных баннеров

На выход:

случайные разные k баннеров

Подход

Интерпретируем сумму весов как отрезок. На отрезке стоят метки: первая метка – вес первого баннера, вторая метка – вес первого + вес второго и так далее. Равновероятно выбираем некоторую точку на отрезке. Тот интервал, в который попала брошенная точка, и будет говорить нам о том, какой баннер показывать.

Прямолинейная реализация – при выборе случайной точки на отрезке каждый раз суммировать веса начиная с первого, пока не превысим случайно выбранное число. Сложность этой реализации $kO(n)$

Был выбран алгоритм, построенный на основе дерева отрезков. Ниже он будет подробно разобран на примере. А затем приведен полный алгоритм отбора k разных баннеров. В самом конце приведено замечание про сдвиг вероятностей показа каждого баннера при увеличении числа k.

Сложность алгоритма:

Каждый раз выбирая баннер мы ищем правильный интервал за $O(\log n)$.

Также стоит учитывать процесс обмена двух элементов, который приводит к обновлению дерева. Обновление дерева производится за $O(\log n)$.

И тогда суммарная сложность алгоритма: k$O(\log n)$

Пример работы алгоритма.

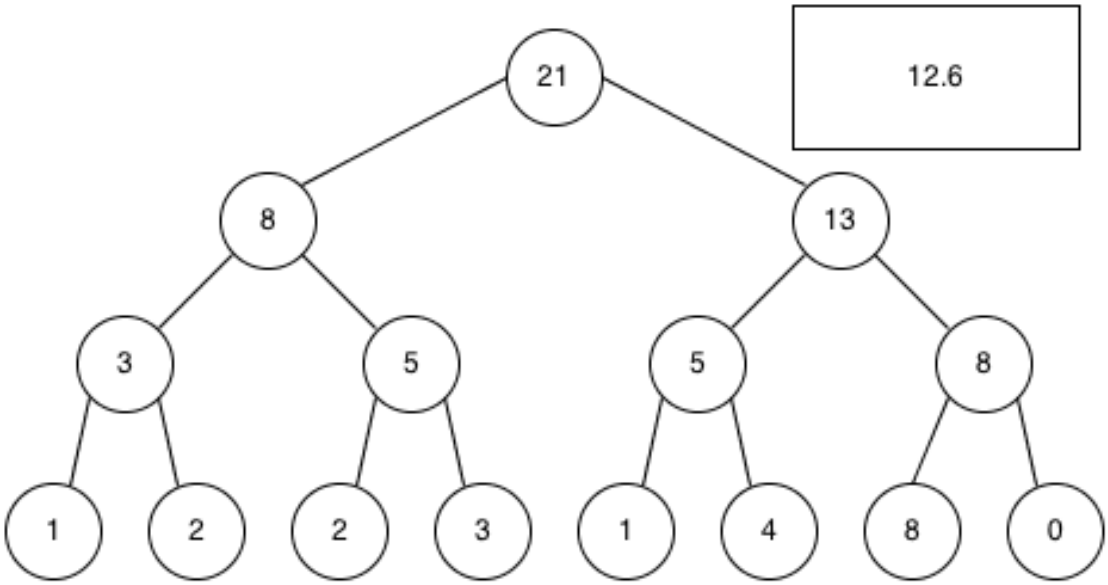
На вход:

[('a', 1), ('b', 2), ('c', 2), ('d', 3), ('e', 1), ('f', 4), ('g', 8)]

Строим дерево отрезков:

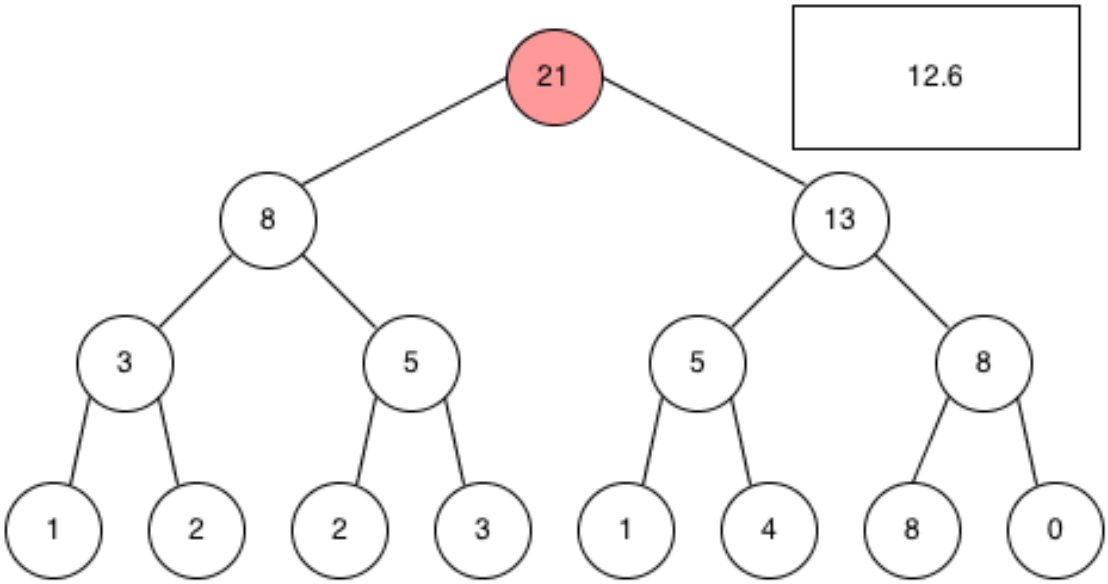
[0, 21, 8, 13, 3, 5, 5, 8, 1, 2, 2, 3, 1, 4, 8, 0]

Выбираем случайно число от 0 до 1. Пускай это будет 0.6. Суммарное количество весов: 21. Значит, нашим генерированным значением считаем generated = 21*0.6 = 12.6



В дереве отрезков нам необходимо найти наименьший отрезок, который начинается с первого элемента, и у которого сумма больше, чем generated.

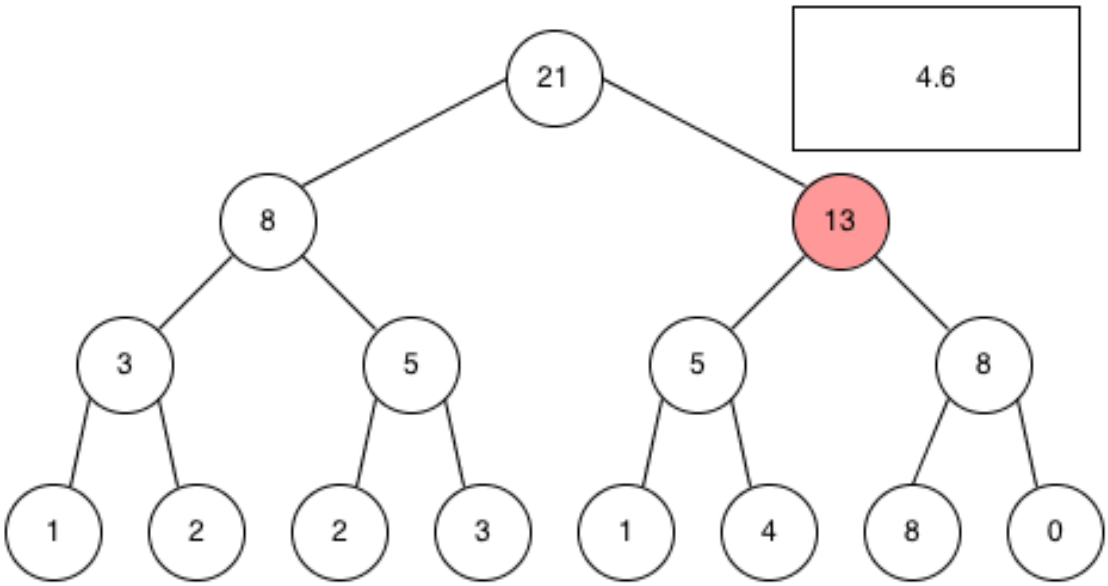
Помечаем корень базовой вершиной.



Смотрим в его левого сына.

8<12.

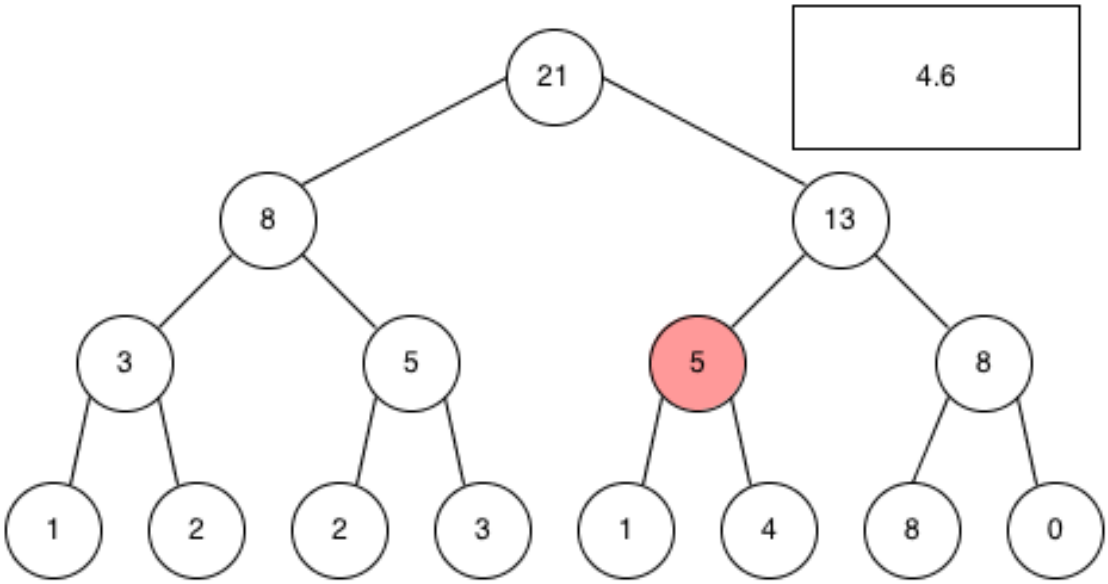
Вычитаем из generated значение в левом сыне и помечаем правого сына базовой вершиной.



Смотрим в левого сына.

5>4.

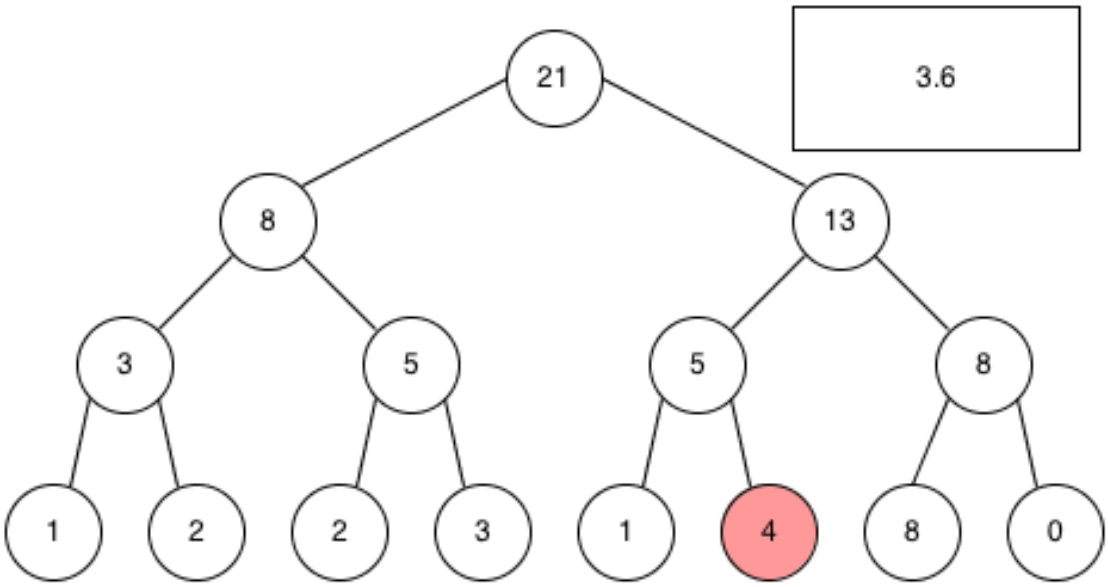
Поэтому помечаем левого сына базовой вершиной



Смотрим на левого сына.

5>1.

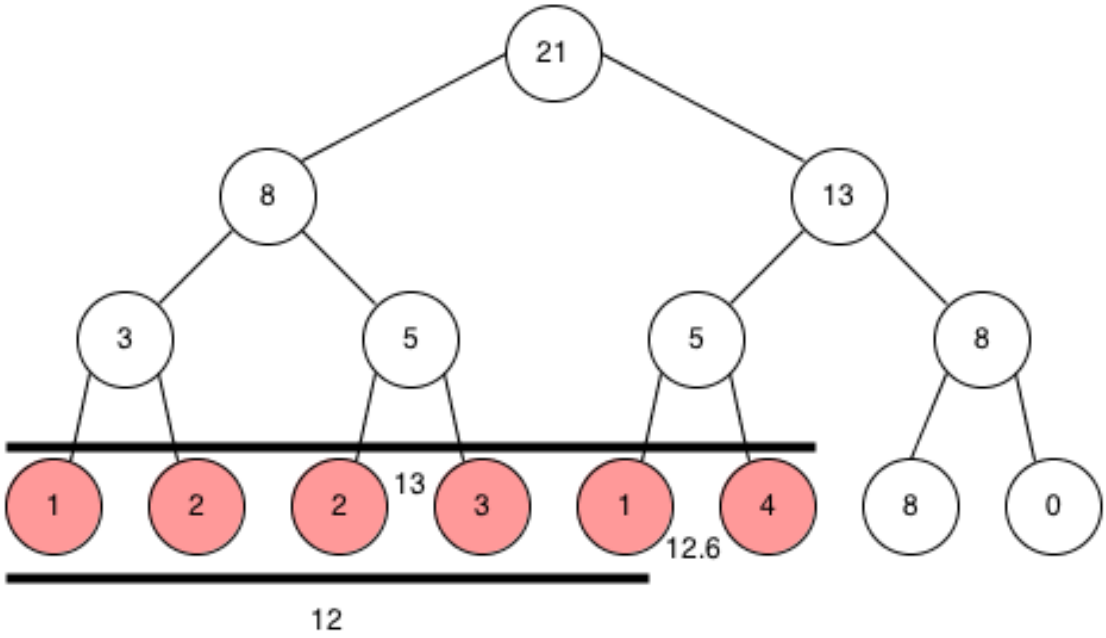
Поэтому вычитаем из generated 1 и помечаем базовой вершиной правого сына.



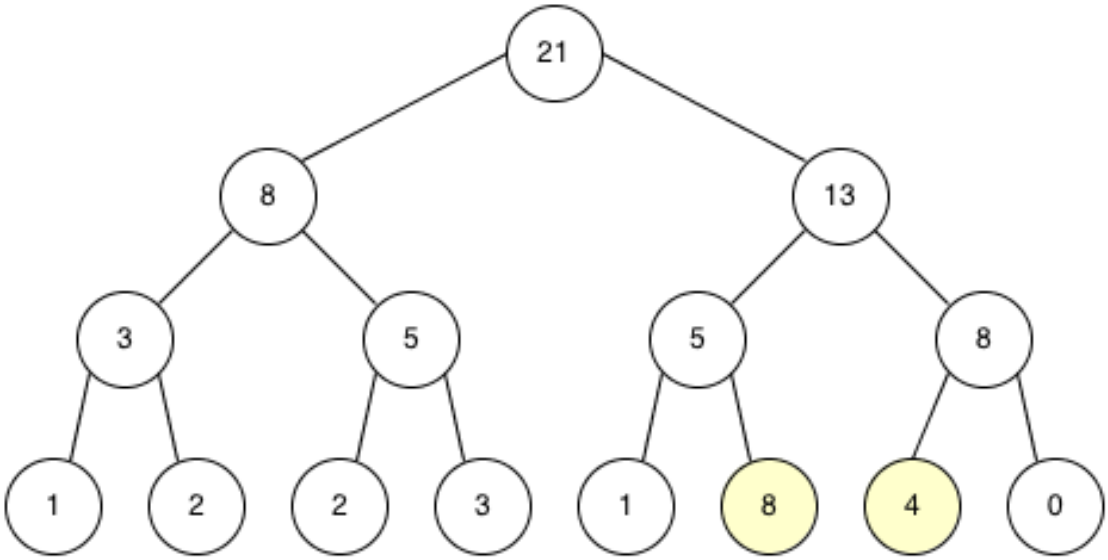
Наша базовая вершина стала листом.

Процесс завершается.

Индекс этой вершины говорит нам о том, что нам нужно показывать 6й баннер

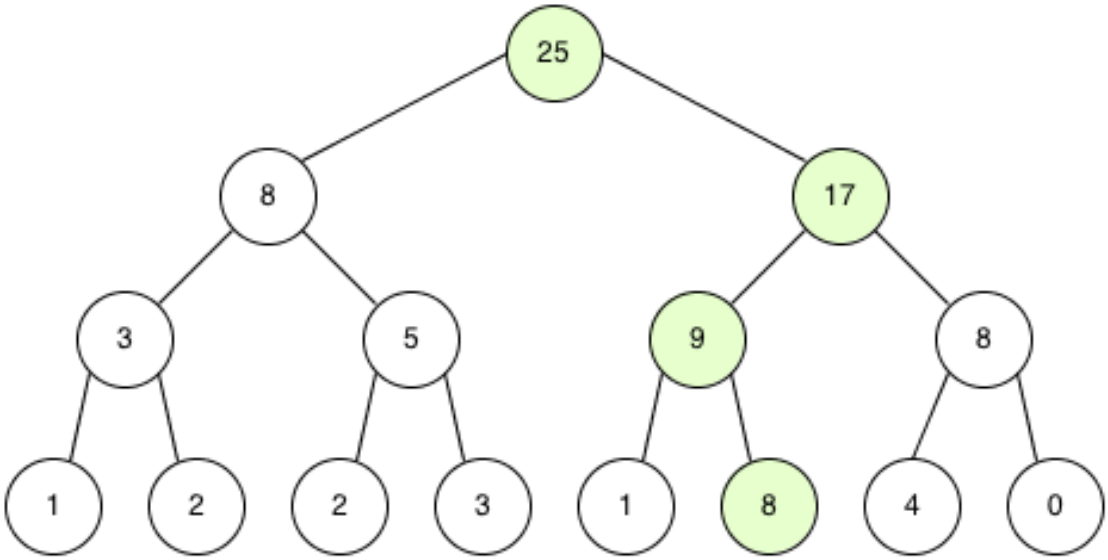


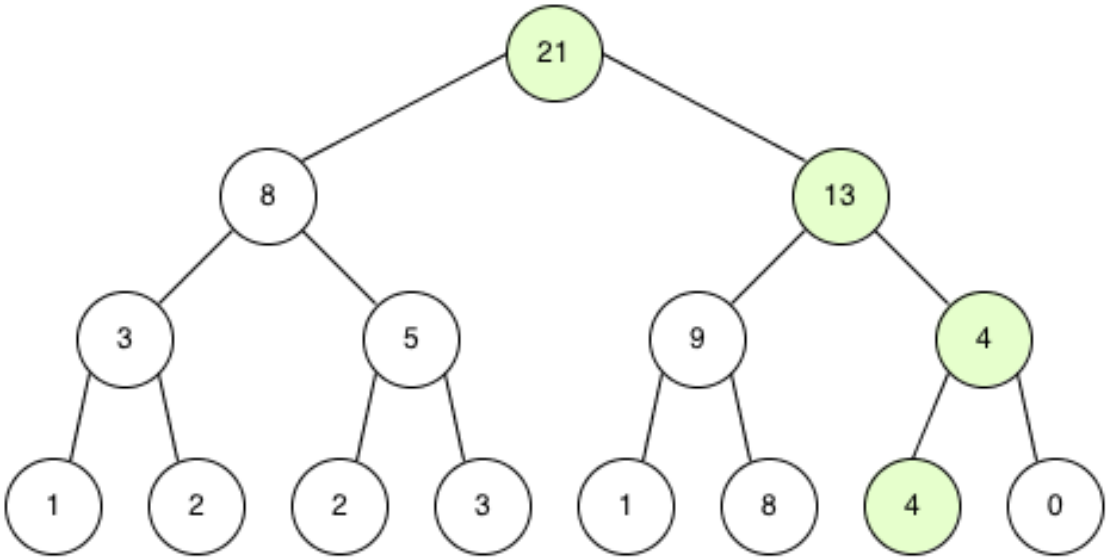
Теперь делаем swap найденного баннера с последним из имеющихся, чтобы при выборе следующего баннера его можно было не рассматривать.



Теперь нужно обновить дерево.

Делаем это для каждой из этих двух вершин.





Теперь мы получили обновленное дерево и можем продолжать выбирать следующий баннер для показа из набора еще не рассмотренных. Из суммарного количества весов для всех баннеров вычитается количество рассмотренных весов: 21-4=17. Генерируется снова число от 0 до 1, пускай 0.5 , умножается на новое суммарное количество весов: 17*0.5 = 8.5, и в снова производится поиск по дереву. Процедура повторяется, пока не покажем все k баннеров.

Алгоритм:

---

1. Строим дерево отрезков для подсчета сумм.
    a. Добиваем до степени 2ки наш массив с весами.

b. Создаем массив по размеру в два раз больший. Во вторую половину записываем наш массив
c. В элементы с индексом k записываем сумму элементов по индексам 2k и 2k+1

2. Генерируем случайное число x из интервала [0,1)
3. Умножаем его на сумму всех рассматриваемых весов
4. Начинаем искать интервал в дереве отрезков для сгенерированного значения
    a. Помечаем корень базовой вершиной
    b. Пока базовая вершина не является листом,
        i. Если значение левого сына базовой вершины меньше, чем сгенерированное значение
            1. Помечаем левого сына базовой вершиной.
            2. Повторяем пункт b
        ii. Иначе
            1. Из сгенерированного значения вычитаем значение левого сына
            2. Помечаем правого сына базовой вершиной
            3. Повторяем с пункта b
    c. Определяем индекс базовой вершины, которая в данный момент уже находится в листе. Это и будет наш интервал.
5. Отображаем баннер, который находится по полученному индексу
6. Меняем баннеры по полученному индексу с последним
7. Обновляем в дереве суммы
8. Повторяем, начиная со второго пункта для нового набора баннеров без последнего элемента. И повторяем, пока не получим k баннеров

Смещение вероятностей

Но хочется при этом отметить, что вероятность появления баннера на экране не будет соответствовать пропорциям.

Пускай у нас есть 4 баннера {A:n1,B:n2,C:n3,D:n4}, из них нам нужно получить 2 разных баннера.

$$P(\textit{показа } A) = P(A \textit{ будет показан в первом блоке})$$
$$+ \sum_{i\epsilon\{B,C,D\}} P(i \textit{ будет показан в первом блоке})$$
$$* P(\textit{ A будет показан в } 1 \textit{ блоке выбирая из баннеров } \{A,B,C,D\} - \{i\})$$

Сравним соотношения вероятностей A и B.

Изначально, когда мы выбираем 1 баннер, соотношение было $P(B) = cP(A)$, то есть n2=cn1

Посчитаем, будет ли теперь такое же соотношение, когда мы выбираем два баннера

$$P(A) = \frac{n1}{n} + \frac{n2}{n}\left(\frac{n1}{n-n2}\right) + \frac{n3}{n}\left(\frac{n1}{n-n3}\right) + \frac{n4}{n}\left(\frac{n1}{n-n4}\right)$$
$$= \frac{n1}{n} + c\frac{n1}{n}\left(\frac{n1}{n-cn1}\right) + \frac{n3}{n}\left(\frac{n1}{n-n3}\right) + \frac{n4}{n}\left(\frac{n1}{n-n4}\right)$$
$$P(B) = \frac{n2}{n} + \frac{n1}{n}\left(\frac{n2}{n-n1}\right) + \frac{n3}{n}\left(\frac{n2}{n-n3}\right) + \frac{n4}{n}\left(\frac{n2}{n-n4}\right)$$
$$= c\frac{n1}{n} + c\frac{n1}{n}\left(\frac{n1}{n-n1}\right) + c\frac{n3}{n}\left(\frac{n1}{n-n3}\right) + c\frac{n4}{n}\left(\frac{n1}{n-n4}\right)$$

Все дроби в P(A), кроме $c\frac{n1}{n}\left(\frac{n1}{n-n1}\right)$, очевидно больше в c раз чем соответствующие им в P(B).

$$\frac{c\frac{n1}{n}\left(\frac{n1}{n-n1}\right)}{c\frac{n1}{n}\left(\frac{n1}{n-cn1}\right)} = \frac{n-cn1}{n-n1} = c - \frac{n(c-1)}{n-n1}$$

Значит при выборе нескольких баннеров соотношение уже выполняться не будет.

Для более общего случая рассмотрено доказательство не было. Но практическим экспериментом было подтверждено, что соотношение вероятностей быть показанными сдвигаются.