

ОПИСАНИЕ

У нас есть табличка на ~500 разных конфет (кол-во постоянно меняется, что-то выходит из ассортимента, что появляется новое, меняется и цена), структура: id, name, price

Мы собираем ориентировочно 50-60 разных подарков, каждый подарок имеет разную стоимость и на наполнение из этих конфет, и у нас проблема с маржинальностью.

КАК СЕЙЧАС ПРОТЕКАЕТ ПРОЦЕСС:

В каждый подарок накидывается определенный список конфет (из тех что есть в табличке, от 10 до 28), и для каждой конфеты задается процент использования конкретно в этом подарке (тем самым условно регулируется кол-во дорогих/дешевых конфет в подарке, для дорогих конфет процент использования ниже, для дешевых – выше, данный механизм позволяет задействовать все конфеты в +- равной степени)

Затем запускается алгоритм, в котором задается **N** (общее кол-во экземпляров конфет в подарке, подарки могут быть разной размерности), передается список выбранных конфет для этого подарка (id, price и % использования), и далее алгоритм генерирует **N** раз случайное число, которое затем проверяет в какой отрезок из процентов попадает, ту конфету и берет.

Таким образом, сейчас собирается подарок на **N** конфет. И вроде бы с наполнением он неплохо справляется, и получаются всегда разные подарки

НО, тут возникает проблема, нужно выдерживать определенную маржинальность при наполнении подарка – **S** (сумма, не более которой нужно набрать конфет). Сейчас маржинальность достигается подбором процента использования конфет в подарок (мучительный процесс), если требуется в определенные периоды повысить или понизить маржинальность начинаются пляски с процентами.

КАК НУЖНО, ЧТОБЫ ПРОТЕКАЛ ПРОЦЕСС:

Мы хотим уйти от ручного указания % (для нас это проблема), упростить алгоритм и сделать точной и понятной для нас маржинальность.

Из исходных данных получаем:

- Подарок вместимостью **N** экземпляров конфет ($10 < N < 100\,000$)
- **S** - максимальная сумма предметов в рюкзаке ($S > 0$)
- **Items** - конфеты с ценой в формате: `[[id, price], [id, price], [id, price], ...]`, где price – float, price > 0

Требуется:

Разработать алгоритм/функцию на любом языке программирования (предпочтительно javascript, python, php), на **вход которому дается:**

N – кол-во экземпляров конфет (вместимость подарка), **S** – **максимальна сумма**, на которую нужно собрать подарок, массив предметов **Items** (например, `[[1, 120.35], [2, 113.45]]`)

на выходе получаем:

сбалансированный *(!) подарок из **N** экземпляров переданных конфет на общую сумму, максимально приближенную к **S**, но не более.

Под сбалансированным подразумеваем, что в подарке будет задействована каждая конфета максимально возможное для нее в этих условиях кол-во раз, то есть не должно быть ситуации (!), когда из переданных 15 конфет,

в подарок используется только 2-3-5 конфет, потому что они оптимальны по цене для достижения заданной N и S, а остальные совсем не используются)

Форматы входных/выходных данных – JSON

Пример входных данных: `{{"N": 54, "S": 123, "ITEMS": [[1, 12.45], [2, 5.41], [3, 1450], [4, 123.54]]}}`

Пример выходных данных (items содержит только ID предметов): `{{"SUM_RESULT": 123.43, "ITEMS": [1, 3, 4, 2, 1, 3, 5, 1, 1, 1, 3, 3, 4, 5, 2]}}`