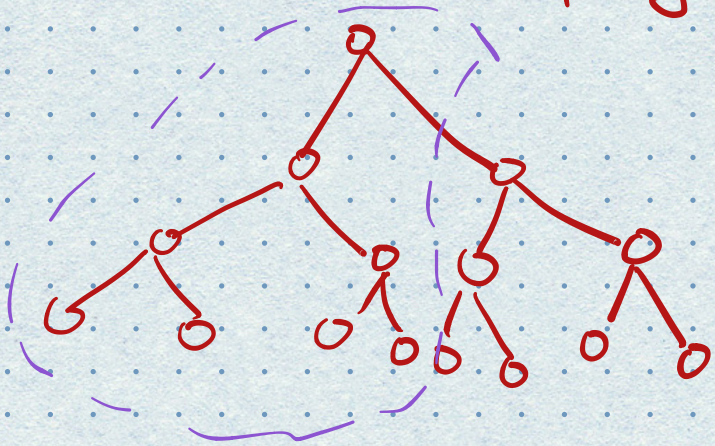
Решаем NP задачи перебором за 2^n

И хотим делать некоторые оптимизации. Которые в общем виде не влияют на асимптотику.

Метод ветвей и границ.



После того, как обработали часть, делаем оценку на оставшуюся. И принимаем решение идти туда или нет.

Отсечение 1

Вес закончился → дальше не идем

Отсечение 2:

Не идем дальше, если сумма стоимостей оставшихся предметов < Best_cost уже имеющийся.

Оптимизация 3:

Решение Колесара.

Если бы задача была непрерывной и мы могли бы брать части вещей, то жадный алгоритм возвращал бы правильное решение при сортировке по удельному весу. (c_i / w_i)

Идея: попробуем сначала самые „лучшие" элементы. Добавим сортировку по удельной стоимости.

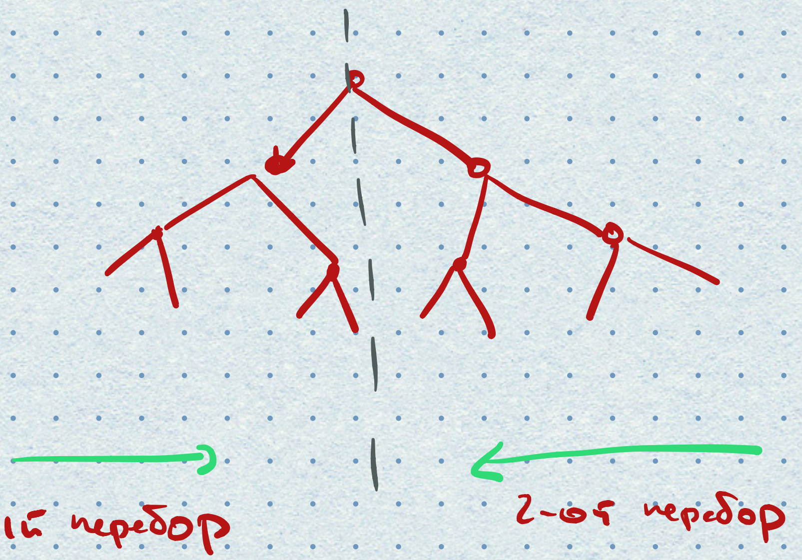
Оптимизация 4:

Попробуем более хорошо искать границу в отсечении 2.

Для оставшихся предметов решим задачу о непрерывном рюкзаке. Решение нашей задачи не лучше. Поэтому если не может заполнить непрерывный рюкзак, то дальше не идем.

Оптимизация 5

Meet in the middle



Если для одного множества можем быстро находить пару из второго мн-ва, тогда сложность будет меньше.

$2^N \times \text{check}$ vs $2^{N/2} \times \underbrace{2^{N/2}}_{\text{можно избавиться}} \times \text{check}$

MiM для рюкзака

- Пусть в левой половине набрали предметов на сумму (W_L, C_L)
- Тогда от правой требуется набор с максимальным C_R при условии $W_R \leqslant W - W_L$
- Все правые множества отсортируем по весу, на префиксах найдём максимальную стоимость
- Ищем бинарным поиском

R - кол-во правых элем.

L - кол-во левых элементов

$R \times 2^R$ - перебор правых

$2^R \times \log(2^R) = R \times 2^R$ - сортировка + префикс. max.

$2^L \times \log(2^R)$ - поиск правого для каждого левого.

Итог: $N \times 2^L + R \times 2^R$ или $N \times 2^{N/2}$