

## 1 Обязательные задачи

1.  $dp[i]$  – мин. число кубов чтобы разложить число  $i$ . Пришли в  $i$ , перебрали  $x$ , обновили  $dp[i]$  через  $dp[i - x * x * x] + 1$ .  $x * x * x \leq n$ , итого  $O(n^{4/3})$ .
2. Выберем произвольную вершину, запустим от неё dfs, насчитаем размеры поддеревьев. Запустим второй dfs, поддерживая во время обхода рамер "наддерева" вершины, то есть до сколько вершин путь лежит через её отца. Теперь просто для каждой вершины проверим, что размер всех её поддеревьев и наддерева не больше  $2n/3$ , и если так, то добавим её в ответ.
3. Давайте скажем, что мы можем занулить весь массив и ответ не станет хуже. Возьмём нулевой элемент и выберем: получали мы профит, когда зануляли его, или нет. Если да, то это значит, что его соседей мы не троагил до того, как занулили его – тогда можно сказать, что его мы занулили первым, а затем сделать линейную дп. Аналогично если мы не получали за него профита, то кто-то из его соседей уже был занулён – давайте выберем, кто, и снова сделаем линейную дп.  
Как делать дп? Пусть мы обрабатываем префикс  $i$ . Тогда мы могли или занулить элементы от  $i - 2$  до  $i$ , и  $dp[i] = zero(i - 1) + dp[i - 3]$ , или нет, и тогда  $dp[i] = dp[i - 1]$ .
4. Если имелось в виду "нестрого выше" и "нестрого ниже", то можно оставить НВП дубов, а все остальные убить, и это наилучший ответ, а если нет, то не умею :(
5.  $2^n$ . Пусть в нашей строке  $s$  есть  $x$  открывающих и  $y$  закрывающих. Тогда строка  $[y * '('] + s + [x * ')']$  правильная.
6. Посчитаем  $dp[i][j]$  – сколько есть последовательностей из  $i$  цифр с суммой цифр  $j$ .  $j \leq 9 * i$ , поэтому всего полей  $n * n * 10$ , а переходов из каждого состояния 10 – по каждой цифре  $k$  из  $dp[i][j]$  в  $dp[i + 1][j + k]$ .  
В конце надо пройти по всем суммам  $s$  и прибавить  $dp[n][s] * dp[n][s]$  к ответу. А ещё можно ускорить это решение, если заметить, что если считать дп назад, а не вперёд, то все переходы из  $dp[i - 1][j - k]$  в  $dp[i][j]$  – это какой-то суффикс массива  $dp[i]$ , то есть можно их считать за  $O(1)$  суфф. суммами.
7. Раз разбиения неупорядоченные, то давайте их упорядочим по возрастанию чисел, на которые бьём.  $dp[i][j][k]$  – кол-во способов разбить число  $i$  на  $j$  слагаемых, чтобы последним было  $k$ .  $dp[i][j][k] = dp[i][j][k + 1] + dp[i - k][j - 1][k]$ , то есть мы или увеличили последнее число на 1, или поставили его в последовательность и перешли дальше.

## 2 Дополнительные задачи

1.

2.

3.  $dp[v][i]$  – кол-во поддеревьев  $v$ , содержащих  $i$  вершин. Если  $u$  – сын  $v$ , то после того, как мы насчитали  $dp[u]$ , мы должны обновить  $dp[v]$ :

для всех  $j$  от 1 до  $k$  и всех  $e$  от 1 до  $k$ :

$dp[v][j + e] += dp[v][j] \cdot dp[u][e]$ .

Это точно работает не дольше  $nk^2$ , но можно оценить лучше: это работает не дольше, чем если бы  $j$  бежали не до  $k$ , а до кол-ва уже обработанных вершин в поддереве  $v$  (то есть в сколько вершин в поддереве  $v$  мы сходили раньше, чем пришли в  $u$ ), и  $e$  не до  $k$ , а до количества детей  $u$ . В таком случае все вместе циклы отработают не дольше чем за количество пар вершин в дереве, а таких  $O(n^2)$ , то есть успех, квадрат есть. Может, это уже работает за  $nk$ , но я не докажу.