Алгоритмы и Структуры Данных ДЗ-11

Гарипов Роман M3138

22.12.2019

Задача №3

Будем считать динамику $dp[mask]$ - сколько подходов мы уже сделали и сколько свободного места осталось, если унесли все предметы, входящие в маску.
База динамики : $dp[0] = \{0, S\}$ Переход : перебираем маску $mask$, для которой уже посчитали динамику, перебираем предмет i который хотим добавить, надо проверить что его нет в маске и если этот предмет весит меньше чем $dp[mask].second$,
то добавляем его к последнему подходу,
$dp[mask + (1 << i)] = \{dp[mask].first, dp[mask].second + w[i]\}$,
иначе $dp[mask + (1 << i)] = \{dp[mask].first + 1, S - w[i]\}$ - в этот подход уже не сможем унести, унесём в следующем, кол-во подходов увеличилось на 1.
Мы рассмотрим все наборы ходов, поэтому динамика посчитается правильно.
Ответом будет $dp[(1 << n) - 1].first$.

Задача №4

Будем доказывать индукцией по высоте дерева

База индукции

Дерево высоты 1 состоящее из одной вершины, она находится на высоте 0

$$\sum_{i=1}^{m} 2^{-d_i} = 1 \leq 1$$

Индукционный переход

Теперь пусть для всех деревьев высотой $h - 1$ и меньше верно. Хотим доказать что верно и для h.
Тогда пусть наша вершина имеет двух сыновей, левый и правый, их поддеревья будут высотой $\leq h - 1$. Тогда для них справедливо следующее :

$$\sum_{i=1}^{size(l)} 2^{-d_{l,i}} \leq 1$$

$$\sum_{j=1}^{size(r)} 2^{-d_{r,j}} \leq 1$$

Сложим эти два неравенства и поделим на два. Необходимо поделить на два так как при соединении этих поддеревьев с нашей вершиной, глубина каждого листа увеличится на 1, следовательно каждое слагаемое уменьшиться вдвое. Запишем что получится :

$$\sum_{i=1}^{size(l)} 2^{-d_{l,i}} + \sum_{j=1}^{size(r)} 2^{-d_{r,j}} \le \frac{1+1}{2} = 1$$

В случае когда у вершины будет всего одного ребёнка - доказательство точно такое же, просто одна из сумм будет равна нулю.

Критерий для равенства 1

$\sum_{i=1}^{m} 2^{-d_i} = 1 \Leftrightarrow$ степень каждой вершины 2 для всех вершин кроме листьев(полное дерево)

Док-во:

$(\Leftarrow)$

Точно такое же доказательство как и для неравенства, только везде будет равенство 1.

$(\Rightarrow)$

Докажем от противного. Пусть есть какое-то неполное дерево(, для которого это неравенство выполняется. Дополним его до полного, добавив нужные вершины. Это добавит нам слагаемых, следовательно сумма получится больше 1. Но мы знаем, что для полного сумма равна 1, получили противоречие, значит дерево обязательно должно быть полным, чтобы выполнялось равенство.

Задача №5

Рассмотрим две вершины : a - вершина в которую мы придём после первого $next$, b - вершина в которую придём после последнего $next$. Так же выделим вершину c - $lca(a, b)$ - наименьший общий предок вершин a и b.
Рассмотрим вершины в левом поддереве вершины c, которые посетит наш алгоритм. Мы посетим все вершины, которые являются результатом вызова очередной операции $next$, т.е. зайдем в поддерево, выйдем из поддерева, пойдём дальше, а так же поднимемся по дереву до c. Суммарно мы пройдёмся по $\mathcal{O}(k)$ вершинам, так как все эти вершины были $next$-ами. Аналогично будет в правом поддереве, но там мы необязательно поднимемся обратно до c.
Получается ассимптотика $\mathcal{O}(\log(n) + k)$, так как дерево сбалансированное,

его высота не более $\mathcal{O}(\log(n))$, и всего вершин, которые являются $next$-ом для кого-то посещено будет $\mathcal{O}(k)$.