1 Задача 3

Будем набирать предметы в подходы последовательно. Создадим массив dp размера 2^n. В каждом $dp[i]$ будем хранить пару значений: количество набранных рюкзаков и оставшееся место в последнем рюкзаке(рюкзаком будем называть набор набранных вещи в подход). Будем говорить что что пара i_1, j_1 оптимальнее пары i_2, j_2 если:

1. $i_1 < i_2$(количество набранных рюкзаков для данной маски предметов меньше $\implies$ набор оптимальнее).
2. $i_1 = i_2$ и $j_1 < j_2$(количество набранных рюкзаков равно, но в последнем осталось больше места $\implies$ набор оптимальнее).

В остальных случаях второй набор оптимальнее.
Будем пересчитывать динамику так: перебираем подмаску, и для каждой подмаски пытаемся добавить в нее каждый возможный предмет, если такое dp уже насчитано, то возьмем более оптимальное, как описано выше. Ответом будет количество рюкзаков в $dp[2^n]$.
Наш алгоритм найдет оптимальное разбиение, так как для каждого dp все его подмаски насчитаны оптимально, предположим что на каком то шаге мы можем как то поменять порядок предметов между рюкзаками и положить последний элемент не в последний рюкзак, но тогда такой случай был учтен когда мы считали dp по маске предметов в этом рюкзаке.
Алгоритм отработает за $\mathcal{O}(2^n n)$, так как мы считаем 2^n dp и для каждого перебираем n предметов которые пытаемся добавить.

2 Задача 4

Докажем первое утверждение по индукции: пусть на обоих подеревьях выполенено условие $\sum\limits_{i=1}^{m} 2^{-d_i} \leq 1$, тогда посчитаем сумму для нашей вершины, она равна сумме результатов на сыновьях деленной на 2, так как для каждой вершины ее глубина увеличилась на 1 $\implies$ каждое слагаемое вида 2^{-d_i} превратилось в $2^{-(d_i+1)} = 2^{-d_i-1} = \frac{1}{2}2^{-d_i}$
Критерий для $\sum\limits_{i=1}^{m} 2^{-d_i} = 1$ — это то, что дерево полное(каждая вершина имеет 0 или 2 ребенка). Докажем это: чтобы сумма была равно 1 нам необходимо, чтобы ответы на сыновьях были так же равны 1, если какая то вершина имеет одного сына, то ее максимальная сумма будет равна $\frac{1}{2}$, так как у единственного ребенка максимальная сумма 1. Тогда если хотя бы у одной вершины сумма не равна 1, то у всех ее предков(а значит и у корня) она будет меньше 1.

3 Задача 5

Рассмотрим начальную(v_1) и конечную(v_k) вершины, пусть между ними есть путь S. Тогда разобьем вершины на внутренние вершины и вершины пути. Внутренние — все вершины правых поддеревьев вершин пути левее lca, и аналогично все вершины левых поддеревьев вершин пути правее lca. Внутренние вершины будут точно посещены, потому, что все они больше v_1 и меньше v_k. Тогда заметим, что внутренние вершины мы посетим не более 3-х раз:

1. На пути в левое поддерево во время поиска другой вершины.
2. Когда ищем саму эту вершину.

3. На пути из правого поддерева — выше.

Таких посещений будет не больше $3k = \mathcal{O}(k)$.
Вершины пути же могут и не входить в множество нужных нам вершин, мы посетим каждую аналогично не более 3-х раз, таких вершин не более $2\log_2 n$, тогда посещений их не более $3 \cdot 2\log_2 n = \mathcal{O}(\log_2 n)$.
Итоговая ассимптотика $\mathcal{O}(\log_2 n + k)$.