对于O(nm^2)的算法假设所有人都会了。f(u,w)=\sum f(v,w1)

而如何去优化时间复杂度

考虑上下界剪枝，对于f(u,w)如何优化，可以优化w，w最大只要装到现在的子树大小

f(v,w1)中w1的下界，只好要让当前size(u)填满后还能继续填，所以为max{w-size(u),0}，而上界只要w＜w1或者size(v)＜w1即可退出，为溢出。

时间复杂度分析：因为(u,v)两点只能在lca处再能计算，而其他处dp已完成，所以时间复杂度O(n^2)