F - Breakdown Editorial by en_translator

Let $\mathrm{dp}[v]$ be the maximum number of operations performable starting from the state where the only piece is placed on vertex v, i.e. the maximum contribution of a piece on vertex v. If this value is found for all $v = 1, 2, \ldots, N$, then the answer is found as $\sum_{v=1}^{N} \mathrm{dp}[v] \times A_v$. Hereinafter, we try to find $\mathrm{dp}[*]$ with Dynamic Programming (DP).

If you remove a piece from a vertex v and pieces are placed on vertices in a set S, then $\mathrm{dp}[u]$ operations originating from each vertex $u \in S$ are performable, for a total of $\sum_{u \in S} \mathrm{dp}[u]$ times. Thus, it is sufficient to choose S that maximizes $\sum_{u \in S} \mathrm{dp}[u]$; in other words,

$$\mathrm{dp}[v] = 1 + \max_{S} \sum_{u \in S} \mathrm{dp}[u]. \tag{1}$$

Here, S takes any (possibly empty) set of vertices adjacent to v such that $\sum_{u \in S} W_u < W_v$.

Since S can only contain vertices u with $W_u < W_v$, the right hand side of equation (1) only consists of vertices u with $W_u < W_v$, so $\mathrm{dp}[v]$ can be successively found for all v in ascending order of W_v.

For a fixed v, the right hand side of equation (1) can be regarded as the following knapsack problem:

> For each of the vertices $u_1, u_2, \ldots, u_{\deg(v)}$ adjacent to v, the **value** of u_i is given as $\mathrm{dp}[u_i]$, and its **cost** as W_{u_i}. Maximize the total value of chosen vertices, subject to the constraint that the total cost adds up to W_v.

This can be solved in $O(\deg(v) \times W_v)$ time with DP.

Since

$$\sum_{v=1}^{N} \deg(v) W_v \leq W_{\max} \sum_{v=1}^{N} \deg(v) = W_{\max} \times 2M,$$

where $W_{\max} \coloneqq \max\{W_1, W_2, \ldots, W_N\}$, one can solve the knapsack problem above for all vertices in a total of $O(M W_{\max})$ time.