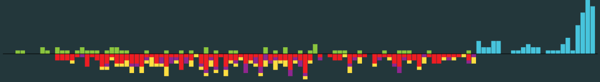
A: Atomic Energy

Problem Author: Jorke de Vlas



Problem

Given are the '*explodification*' rules for an atom with a certain amount of neutrons:

- An atom with $k \leq n$ neutrons will be converted into a_k units of energy.
- An atom with $k > n$ will be decomposed into parts $i, j \geq 1$ with $i + j = k$, which are then recursively *explodificated*.

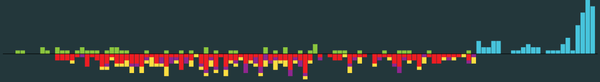
Given an atom with a fixed number of neutrons, what is the minimum energy released?

Observations

Since the decomposition is arbitrary, we have to assume the worst case – for $k > n$ define:

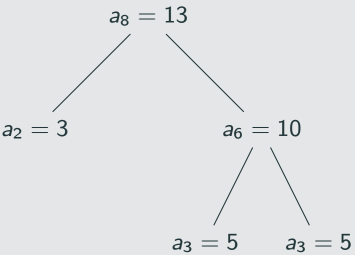
$$a_k := \min_{1 \leq i \leq k-1} a_i + a_{k-i}.$$

There are upto 10^5 queries with k upto 10^9, so we cannot naively compute all values a_i upto this maximum. Naive computation requires $O(k^2)$ time for the first k values.

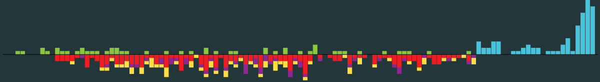


Observation 1

Our first crucial observation is that optimal solutions have a recursive structure. We can write any explodification sequence as a binary tree. This is the first sample, $k = 8$:



Recall this sample had $a_{1,\dots,4} = \{2, 3, 5, 7\}$.



Observation 1

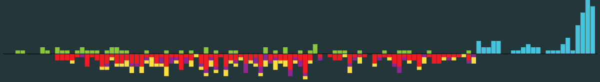
For a given query k, imagine recursively following the decomposition $a_k = a_i + a_{k-i}$ until we end up with a decomposition:

$$a_k = \sum_{j=1}^{m} a_{i_j} \quad \text{subj. to} \quad k = \sum_{j=1}^{m} i_j, \text{ with } i_j \in \{1, \ldots, n\}.$$

So the leaves of the decomposition tree are a collection of indices i_j that sum to k. Is any decomposition (i_j) satisfying the right hand side realizable?

No – to actually construct this explodification sequence we need to end with some a_x, a_y with $x + y > n$. If $x + y \leq n$, there is no guarantee that $a_{x+y} = a_x + a_y$. (Example: for $n \gg 1$, a sequence of all a_1's is generally impossible.)

A sequence is *realizable* if it contains two x, y with $x + y > n$. After that, we can 'add' new atoms a_{i_j} inductively to construct the explodification tree. In fact any 'prefix' of such a sequence is optimal.

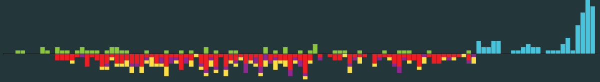


Faster computation

Now we can improve the computation of the first k values from $O(k^2)$ to $O(nk)$:

$$a_k = \min_{1 \le i \le n} a_i + a_{k-i}.$$

Of course this is still not fast enough with k upto 10^9.



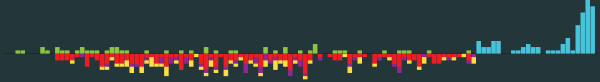
Observation 2

Let $m \in \{1, \ldots, n\}$ minimize a_m/m. When a query k is large enough, most of the terms in the decomposition will be a_m. Indeed, if after removing the two distinguished values a_x, a_y from the sequence we still have m or more values in the tree that are not a_m, by the pigeonhole principle there must be a subset of them that have indices that sum up to a multiple of m, and we can replace them by a_m's to get a decomposition that is not worse.

Hence, any decomposition can be written in such a way that there are at most $m+1$ terms that are not a_m. In fact we can rearrange the sequence to have these terms in the front, and then fill in the gap with a_m-terms.

A: Atomic Energy

Problem Author: Jorke de Vlas



Full solution

Let m minimize a_i/i over all $i \in \{1, \ldots, n\}$, and use the $O(nk)$ algorithm from earlier to construct the first $(m+1)n$ terms in time $O(n^3)$.

For each query k, find the smallest $j \geq 0$ such that $k - jm \in \{1, \ldots, (m+1)n\}$, and output with $a_{k-jm} + j \cdot a_m$.

Final runtime $O(n^3 + q)$. Efficient implementations of e.g. $O(n^4 + q)$ could also work.

Statistics: 421 submissions, 51 + ? accepted