A Difficult(y) Choice (books)

BY SERGEJS MEĻŅIKS (LATVIA)

Subtask 1. $S = N$, $200 \le N \le 1\,000$, $K = 3$

For this subtask you can skim all books and then search for a suitable set of 3 to answer. This can easily be done in $O(N^2 \log N)$.

Subtask 2. $S = N$, $200 \le N \le 25\,000$

Again, we first skim through all books. You can see that if there is an answer, there are K consecutive books that suit or one book with difficulty greater A and $K - 1$ books easier than A. Namely, if the first $K - 1$ books and the smallest $x_i > A$ do not form a solution, then x_i and any higher difficulty books are not part of the answer. In this case, if there is any solution, there is also a solution with consecutive books: as all remaining books have difficulty at most A, passing from one sequence of consecutive books to the next one increases the sum by at most A, so we can't jump from $< A$ to $> 2A$.

In linear time you can check whether one of these two possibilities gives a suitable answer.

Subtask 3. $S \ge 200$, $x_{i+1} - x_i \le A/K$ for all $1 \le i \le N - 1$

In this case there always exists either a consecutive answer or no answer. You can binary search for this sequence with $K \log(N)$ skims.

Subtask 4. $S \ge 200$, $x_{i+1} - x_i \le A$ for all $1 \le i \le N - 1$

We notice, that if all but one books are fixed, changing the remaining book by one will change the sum by at most A and thus we can't jump from below an acceptable difficulty to above an acceptable difficulty. We now start with the easiest K books and then one by one (from the highest difficulty to the lowest) use a binary search to find the highest difficulty book which doesn't make the total difficulty too high.

Subtask 5. $S \ge 200$

This is the same as the solution for Subtask 2, except that we now use binary search to find the smallest $x_i > A$ as well as the smallest sum $\ge A$ of consecutive elements.

Subtask 6. $S \ge 40$, $x_{i+1} - x_i \le A$ for all $1 \le i \le N - 1$

We notice, that if all but one books are fixed, changing the remaining book by one will change the sum by at most A and thus we can't jump from below an acceptable difficulty to above an acceptable difficulty. We now start with the easiest K books and one by one change them to the highest possible difficulty books. Whenever the total difficulty jumps from below the acceptable difficulty to above the acceptable difficulty we can now use a binary search to find a replacement for that last book which makes the total difficulty acceptable.

Subtask 7. $S \geq 40$

The following yields a total query complexity of $O(\log(N) + K)$ queries:

If the first $K - 1$ books and the smallest $x_i > A$ do not form a solution, then x_i and any higher difficulty books are not part of the answer. Let the remaining numbers be called $x_1, \ldots, x_{i-1}$. Assume for ease of exposition that $K < 2i$, so that $x_1 < \ldots < x_K < x_{i-K} < \ldots < x_{i-1}$. It can be shown that if the solution exists, it will be one of the following:

- $x_1, \ldots, x_K$
- $x_2, \ldots x_K, x_{i-K}$
- …
- $x_K, x_{i-K}, \ldots x_{i-2}$
- $x_{i-K}, \ldots x_{i-1}$;

namely, whenever we pass from one collection to the next one, the total difficulty can change by at most A, so we cannot jump over the desired interval.