**Analysis of the problem H – Railway Station**

The railway station is a stack. The train arriving from the direction *A* has *n* ≤1000 coaches numbered in increasing order 1, 2, ..., *n*. And the coaches leaving in the direction *B* are numbered as a permutation for 1, 2, ..., *n*. The permutation is implemented by stack operations.

For each test case, the permutation is simulated. That is, coaches 1, 2, ..., *n* in the direction *A* are pushed into the stack and compared with the elements in the permutation one by one, to determine whether the permutation is valid or not.

1. If the current element in the direction *A* (that is, the element will be pushed into the stack) is same as the current element in the permutation; then the current element in the direction *A* will be pushed into the stack and popped from the stack directly. Next elements in the direction *A* and in the permutation become the current elements.
2. If the element at the top of stack is same as the current element in the permutation, then the element will be popped from the stack. The next element in the permutation becomes the current element in the permutation.
3. Otherwise the current element in the direction *A* is pushed into the stack. The next element in the direction *A* becomes the current element in the direction *A*.

Repeat above steps. If *n* elements in the permutation can be popped from the stack, then the permutation is valid; otherwise the permutation isn’t valid. Its time complexity is O(*n*).

**3. The data structure and the algorithm to be used.**

data structure: stack

algorithm: stack