**Task 4. «Birch alley»**

Depending on the total number of birches *V (V = N + M)*, this problem has a number of partial and complete solutions. Consider them sequentially, " from simple to complex."

*The first partial solution*, which has asymptotic complexity О(*V*4) and allows to get an answer for *1 ≤ V ≤ 50*, is based on a complete iteration and is as follows. Consider the points *i*1 and *j*1 (*j*1≥*i*1) on the left side of the alley and the points *i*2 and  *j*2 (*j*2≥*i*2) on the right side. For each four of these points, calculate the perimeter of the "quadrilateral" with vertices at these points, Compare these perimeters with the length of the tape, and among the quadrilaterals whose perimeters are not greater than the length of the tape, find the one on the border of which there are most birches (the number of birches on the border is (*j*1–*i*1+1)+(*j*2–*i*2+1)).

Note that here a quadrilateral can degenerate into a triangle, and even into a segment. In order not to miss these cases, equality of birch numbers is allowed in inequalities (*j*1≥*i*1, *j*2≥*i*2).  In the future we will take this into account.

*The second partial solution*, which has asymptotic complexity О(*V*3), yields the answer for 1 ≤ *V* ≤ 500 and is as follows. Let's choose some *i*1, *j*1, *i*2 (notations are similar to the previous partial solution). Note that for these three selected values, we are actually interested in only one value *j*2 – the maximum of those values for which the perimeter of the quadrilateral does not exceed the length of the ribbon. Let us now consider the points of *i*1, (*j*1+1), *i*2. Note that the value of *j*2’ for a new triple of numbers cannot be greater than  *j*2, that is, with the growth of *j*1, the value of *j*2 does not increase. Therefore, to find the desired value of *j*2’, you can reduce it, starting with the value of *j*2 where we left off in the previous step, until the perimeter is less than or equal to the length of the tape.

This technique uses the idea of "two pointers": in this case, for fixed *i*1 and *i*2, "pointers" *j*1 and *j*2 run the entire range of values once (*j*1 – in ascending order, *j*2 – in descending order), that is, for each pair (*i*1 and *i*2), the time of the algorithm is linear. If, instead of the idea of two pointers, we use binary search to find *j*2, we can obtain a solution having asymptotic complexity *V*3·log *V*.

Along with the described *partial solutions*, there are several ways to apply *complete solutions*. The *first way* is based on the use of binary answer search in conjunction with the interval tree and has asymptotic complexity О(*V*2×log2*V*). We will describe how to check whether it is possible to protect a certain fixed number of birch trees S with a given tape.

Let's fix the birches *i*1 and *i*2 as in the previous solutions, and consider how to check for the time of the order log *V* whether there is a quadrilateral with a perimeter not exceeding *L*, and containing *S* birches. Note that for all pairs of interest (*j*1,*j*2) the value of the sum (*j*1+*j*2) is the same, since (*j*1+*j*2–*i*1–*i*2+2)=*S*. Among them, we are interested in the pair for which the perimeter of the quadrilateral with vertices *i1, j1, j2, i2* is minimal. But for the same pair, the perimeter of a quadrilateral with vertices 1, *j1, j2*, 1 is also minimal among all quadrilaterals for which (*j*1+*j*2) is equal to a fixed value (*S* + *i*1 + *i*2 – 2). It is important that these values are no longer dependent on *i*1 and *i*2, they can be counted once and then used. But in fact we do not need all quadrilaterals, but only those for which *j*1≥*i*1and *j*2≥*i*2. To find among them a quadrangle with a minimum perimeter for the time of the order log*V*, we modify the pre-calculation as follows.

 Let *a*[*S*][*k*]  be the perimeter of a quadrilateral with vertices: 1, *k, (S – k),* 1, that is, the array string *a*[*S*] contains the perimeters of all quadrilaterals spanning exactly *S*birches. For each line of the array *a*, we construct a tree of segments for the function min, that is, for a fixed number of birches *S,* we can calculate min{*a*[*S*][*x*], …, *a*[*S*][*y*]} with logarithmic asymptotic complexity, which is required. Now to find the required quadrangle it is enough to answer the query:

min{*a*[*S*+*i*1+*i*2 – 2][*j*1],…,*a*[*S*+*i*1+*i*2–2][*S*+*i*1+*i*2–2–*j*2].

*The second way* of the complete solution has asymptotic complexity О(*V*2·log *V*). This complexity can be achieved by replacing the interval tree with sparse tables in the previous solution. In this case, the memory costs will increase to *V*2×log*V*.

*The second way* of the complete solution can be improved from memory, if you pay attention to the following fact. We will iterate over the pairs (*i*1,*j*1) in order of increasing the sum (*i*1+*j*1), and for equal sums – in order of increasing *i*1. For each such sum, with the growth of *i*1, both ends of the segment of valid (for given *i*1) values of *j*2 will decrease. Then you can use a queue to store this segment with support of adding to the end, removing from the beginning, and searching in at least O(1). In this case, we need only linear additional memory to store the queue, the length of which at any given time will not exceed N.