

NOI '22 P6 - Quadratic Integer Program

Time limit: 2.0s Memory limit: 1G

Task description

In this problem, you need to solve a well-known NP problem - the quadratic integer programming problem.

Quadratic integer programming problems have variables: you need to give a length n sequence of integers $(x[1], x[2], \dots, x[n])$ that satisfies all the conditions below.

Quadratic integer programming problems have constraints: the sequence of integers you give needs to satisfy the following two types of constraints:

1. Given a positive integer k ($3 \leq k \leq 5$) and n intervals $[l[i], r[i]]$ ($1 \leq i \leq n$), where $1 \leq l[i] \leq r[i] \leq k$, the sequence you give needs to satisfy $l[i] \leq x[i] \leq r[i]$ for all $1 \leq i \leq n$.
2. Given m triples $(p[j], q[j], b[j])$, the sequence you give needs to satisfy $|x[p[j]] - x[q[j]]| \leq b[j]$ for all $1 \leq j \leq m$.

The quadratic integer programming problem has an objective function: you are given given $k - 2$ weight parameters $v[2], v[3], \dots, v[k - 1]$ (note that the indices are from 2 to $k - 1$). Let $c[i]$ be the number of elements in the sequence whose value is i , and G be the number of pairs $1 \leq i, j \leq n$ such that $|p[i] - p[j]| \leq 1$ (note that when $i \neq j$, (i, j) and (j, i) are not the same). The weight of a sequence $x[1], x[2], \dots, x[n]$ is:

$$W(x[1], x[2], \dots, x[n]) = 10^6 G + \sum_{i=2}^{k-1} c[i] v[i]$$

Your sequence needs to maximize its weight while satisfying the above two constraints.

Quadratic integer programming problems do not necessarily require multiple queries, but we will give q queries, each query giving different weight parameters $v[2], v[3], \dots, v[k - 1]$. For each query, you need find a sequence of maximum weight that satisfies the constraints. To reduce the output, you only need to output the weight of this sequence. The data guarantees that at least one sequence that satisfies the above conditions exists.

Input Specification

There are multiple sets of test data for this question. The first line contains a non-negative integer C and a positive integer T , which represent the test case number and the number of data sets, respectively. $C = 0$ indicates that this set of data is a sample.

For each test case:

- The first row contains four integers k, n, m, q , describing the sequence range, sequence length, the number of constraints between variables, and the number of queries.
- The next n lines each contain two integers $l[i], r[i]$, describing the value range corresponding to each element in the sequence.
- The next m lines contain three integers $p[j], q[j], b[j]$ each, describing the constraints between a pair of variable.
- Each of the next q lines contains $k - 2$ non-negative integers $v[2], v[3], \dots, v[k - 1]$ describing a set of query weight parameters.

Output Specification

For each set of queries for each set of data, output a row of integers representing the maximum weight of the sequence.

Samples

Sample inputs and outputs can be found [here](#).

Sample 1

See qip/qip1.in and qip/qip1.ans in the player directory. This example satisfies the properties of test point 1 in the data range.

Explanation of Sample 1

In the first test data, the optimal sequences corresponding to the two sets of queries are (1, 2, 2, 1, 3), with $c_2 = 2$, $G = 21$.

Example 2

See qip/qip2.in and qip/qip2.ans in the player directory.

This example satisfies the properties of test point 3 in the data range.

Explanation of Sample 2

An optimal sequence corresponding to the two sets of queries in the first test data is (4, 4, 3, 3) and (4, 3, 2, 2), respectively.

Sample 3

See qip/qip3.in and qip/qip3.ans in the player directory. This example satisfies the properties of test point 5 in the data range.

Explanation of Sample 3

An optimal sequence corresponding to the three groups of queries in the first test data is (3, 3, 3, 3, 3), (2, 2, 3, 3, 2) and (3, 2, 4, 4, 2).

Sample 4

See qip/qip4.in and qip/qip4.ans in the player directory. This sample satisfies the properties of test point 2 in the data range.

Sample 5

See qip/qip5.in and qip/qip5.ans in the player directory. This sample satisfies the properties of test point 4 in the data range.

Sample 6

See qip/qip6.in and qip/qip6.ans in the player directory. This sample satisfies the properties of test point 8 in the data range.

Sample 7

See qip/qip7.in and qip/qip7.ans in the player directory. This sample satisfies the properties of test point 14 in the data range.

Sample 8

See qip/qip8.in and qip/qip8.ans in the player directory. This sample satisfies the properties of test point 17 in the data range.

Subtasks

Assume $\sum q$ is the sum of q of all test data in a single test point. For all test points,

- $1 \leq T \leq 600$,
- In the i ($1 \leq i \leq T$) th test data, $1 \leq n \leq \max(T/i, 2\log_2 T)$,
- $3 \leq k \leq 5, 0 \leq m \leq 3n, 1 \leq q, \sum q \leq 3 \times 10^5$,
- $1 \leq l[i] \leq r[i] \leq k$,
- $1 \leq p[j], q[j] \leq n, 0 \leq b[j] < k$,
- $0 \leq v[2], \dots, v[k-1] \leq 10^{12}$.

Case	$T \leq$	$k =$	$\sum q \leq$	Special Property	Points
1	10	3	200	None	4
2	600		3×10^5		6
3	10	4	200		4
4	600		3×10^5		6

5	10	5	300	A	5	
6	15		500		4	
7	25		750		4	
8	50		1000		6	
9	80		1500		6	
10	120		2000		5	
11	200		8000		3	
12	400		3×10^4		4	
13	600		2×10^5		5	
14	200		8000		B	3
15	400		3×10^4			4
16	600		2×10^5	4		
17	120		10^5	C	4	
18	150		2×10^5		5	
19	180		3×10^5		5	
20	300		5×10^4	None	5	
21	450		10^5		4	
22	600		3×10^5		4	

- Special property A: $m = 0$.
- Special property B: $m \leq 10$, the sum of m of all test data in a single test point does not exceed 200.
- Special property C: The data is randomly generated. Specifically, when generating each set of test data in the test point, the parameters k, n, m, q and k non-negative integers $p[0], p[1], p[2], \dots, p[k-1]$ are given to ensure that $p[k-1] \neq 0$, then the following rules are used to generate this data:
 - For $1 \leq i \leq n$, independent uniform random generation of $x, y \in [1, k]$, then $l[i] = \min(x, y), r[i] = \max(x, y)$;
 - Keep generating triples as follows until there are m triples:
 - Independently and uniformly randomly generate $u, v \in [1, n]$;
 - Randomly generate w with p as weight (for $0 \leq i \leq k-1, w = i$ with probability $\frac{p[i]}{p[0]+p[1]+\dots+p[k-1]}$);
 - If there is no sequence $(x[1], x[2], \dots, x[n])$ after adding (u, v, w) to the original triple set, the current triple is discarded, otherwise the current triple is added.
- $v[2], \dots, v[k-1]$ of each set of queries are independently and uniformly generated randomly within $[0, 10^{12}]$.