Graph

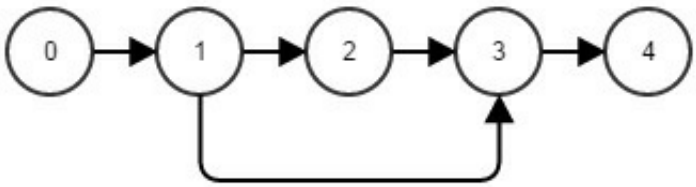
You are given a directed graph with N vertices and M directed edges. Vertices are numbered with integers from 0 to $N-1$. i-th edge starts from vertex $U[i]$ and leads to vertex $V[i]$.

For given two different vertices S and T calculate number of vertices that lay on each path from S to T (do not count S and T).

Example

In this example $N = 5$, $M = 5$, $S = 0$, $T = 4$.

	0	1	2	3	4
U	0	1	2	3	1
V	1	2	3	4	3



Vertices number 1 and 3 are definetely on each path. But we can avoid vertex number 2. So answer is 2.

Task

Please, write a program that calculates number of vertices that lay on each path from S to T excluding S and T. You need to implement the following function.

- `calculate(N, M, S, T, U, V);`
  - `N`: number of vertices.
  - `M`: amount of edges.
  - `S`: starting vertex.
  - `T`: finishing vertex.
  - `U`: array of length M, starting vertex for each edge. $0 \le U[i] < N$, $0 \le i < N$.

- V: array of length M, finishing vertex for each edge. $0 \leq V[i] < N$, $0 \leq i < N$.
- The function should return the answer to the task.

Subtasks

subtask	points	N	M
1	20	$1 \leq N \leq 500$	$0 \leq M \leq 1,000$
2	30	$1 \leq N \leq 5,000$	$0 \leq M \leq 10,000$
3	50	$1 \leq N \leq 100,000$	$0 \leq M \leq 200,000$

Implementation details

You have to submit exactly one file, called `graph.c`, `graph.cpp`, `graph.pas` or `graph.java`. This file should implement the subprogram described above as a function or method, using the following signatures.

C/C++ program (include `graph.h` at the top of the source file)

```
int calculate(int N, int M, int S, int T, int U[], int V[]);
```

Pascal programs (implement the described method in unit `graph`)

```
function calculate(N, M, S, T : longint; var U, V : array of longint) : l
```

Java programs (implement the described method in public class `graph`)

```
public int calculate(int N, int M, int S, int T, int[] U, int[] V);
```

Sample grader

The sample grader reads the input in the following format:

- line 1: `N M S T`
- line $2 + i$: ($0 \leq i < M$): `U[i] V[i]`

The sample grader will print out your answer.