

---

CSC 284/484 Advanced Algorithms - applied homework 1  
due: February 12th, 11:59pm EST

---

Grading:

- 284: 1 problem solved = A
- 484: 2 problems solved = A, 1 problem solved = B

This homework has different rules than the theoretical homework, most importantly, do not discuss any aspect of the applied problem with anybody else (except me), do not search for a solution online, do not use any written material when writing any part of the code (for example, no copy-paste, no open textbook when writing code, no reediting of an old source file from an old project, etc).

PROBLEM 5 - implement any max-flow algorithm

---

Implement any  $s$ - $t$ -max-flow algorithm.

INPUT FORMAT: The first line contains  $k$ , the number of problems. Then descriptions of the problems follow. The first line contains  $n$  (the number of vertices) and  $m$  (the number of edges). The second line contains  $s, t \in \{1, \dots, n\}$  (the source and the sink). Then  $m$  lines follow. Each line contains three integers  $u, v, c$ , where  $u, v \in \{1, \dots, n\}$  and  $1 \leq c \leq 1,000,000$ ; this means that there is edge  $(u, v)$  with capacity  $c$  in the graph. We have  $2 \leq n \leq 1,000$  and  $1 \leq m \leq 10,000$ .

OUTPUT FORMAT: The output contains one line for each problem—the value of the maximum  $s$ - $t$  flow.

EXAMPLE INPUT:

```
2
2 1
1 2
1 2 10
2 1
2 1
1 2 10
```

EXAMPLE OUTPUT:

```
10
0
```

## PROBLEM 6

We are given a directed graph  $G = (V, E)$ , two vertices  $s$  and  $t$  and a positive integer  $K$ . We want to find  $s$ - $t$ -walks  $P_1, \dots, P_K$  such that no two walks have the same  $i$ -th edge (for any  $i$ ) and the length of the longest walk is minimized.

(We describe the problem more formally. Let walk  $P_j$  be  $v_{j,0}, \dots, v_{j,\ell_j}$  where  $v_{j,0} = s$  and  $v_{j,\ell_j} = t$ . We want to minimize  $\max_{j \in [K]} \ell_j$ . The condition on the walks is: for any distinct  $j, h \in [K]$  and any  $i \in [\min\{\ell_j, \ell_h\}]$  we have  $(v_{j,i-1}, v_{j,i}) \neq (v_{h,i-1}, v_{h,i})$ .)

INPUT FORMAT: The first line contains the number of test cases. Each test case is described on several lines. The first line contains  $n$  (the number of vertices) and  $m$  (the number of edges). The second line contains three numbers  $s, t, K \in \{1, \dots, n\}$ .

Then  $m$  lines follow. Each line contains two integers  $u, v$ , where  $u, v \in \{1, \dots, n\}$ ; this means that there is edge  $(u, v)$  in the graph. We have  $2 \leq n \leq 100$  and  $1 \leq m \leq 1,000$ .

OUTPUT FORMAT: The output contains one line for each test case. If no collection of valid walks exists then the line should contain NO. Otherwise the line contains a single integer, the length of the longest walk in the optimal solution.

EXAMPLE INPUT:

```
2
2 1
1 2 2
1 2
5 6
1 5 2
1 2
1 3
3 4
2 4
4 2
4 5
```

EXAMPLE OUTPUT:

```
NO
5
```

(For the second problem in the example we can take walks  $1, 2, 4, 2, 4, 5$  and  $1, 3, 4, 5$ ; note that pair of walks  $1, 2, 4, 5$  and  $1, 3, 4, 5$  is not a valid solution (they have the same 3rd edge).)