# Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

**Problem:** *Buffering.*

A long awaited movie will be finally been made available on Graphflex, a popular streaming platform.

You are looking forward to watch it but your internet provider, Busycomm, has very slow and inconsistent speeds. To avoid interruptions you decided to let the movie buffer completely before starting to watch it.

Through not quite legal means, you also managed to find i) a map of the $n$ network hosts of Busycomm, indexed from 0 to $n-1$, and of the $m$ links between them, and ii) an internal document that reports, for each (birectional) link $(u, v)$ and for each hour of the day $h = 0, \dots, 23$, the speed $b_h(u, v)$ across $(u, v)$ in bytes per hour. Your router communicates directly with host 0 with a high-speed link, while the $k$ hosts $n-k-1, n-k, \dots, n-1$ can directly route traffic to the internet backbone. You can assume that the bandwidth between your router and host 0, and between the backbone and the Graphflex servers is essentially unlimited. You can also assume that an optimal load-balancing algorithm is employed that allows different parts of the movie to be simultaneously sent over different links.

The movie has a size of $B$ bytes and you will start buffering it as soon as it is released, i.e., at midnight of the release day. Design an algorithm that computes how many hours will take for the movie to buffer completely.

**Input.** The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number $T$ of test-cases. The first line of each test-case contains the four integers $n$, $m$, $B$ and $k$. Each of the following $m$ lines describes one link $(u, v)$ end contains the 26 integers $u, v, b_0(u, v), b_1(u, v), \dots, b_{23}(u, v)$.

**Output.** The output consists of $T$ lines, each containing a single character. The $i$-th line is the answer to the $i$-th test-case and contains the minimum number of hours $H$ needed to send all $B$ bytes of the movie from Graphflex server to your router (round $H$ up to the next hour). Notice that the download might take more than 24 hours.

**Assumptions.** $1 \le T \le 10$; $1 \le n \le 2^{14}$; $1 \le B \le 2^{20}$; $1 \le k \le n-1$;
All $b_j(u, v)$ are between 0 and $2^{10}$. The buffering can always be completed.

**Example.**

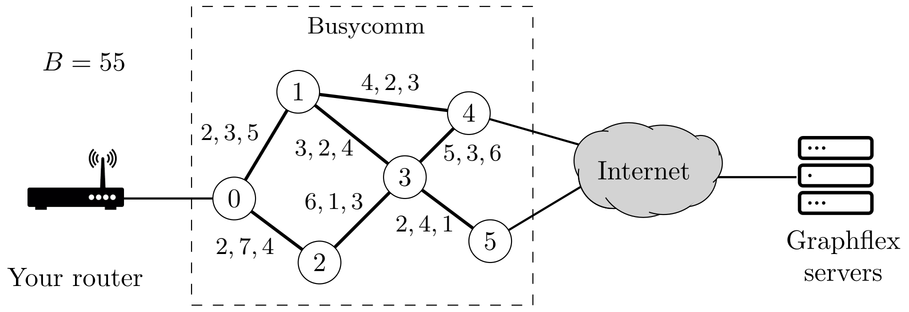

Figure 1: An example of the map of the network hosts of Busicomm, along with their bandwidths. For simplicity, only the speeds $b_1(u, v)$, $b_2(u, v)$, and $b_3(u, v)$ corresponding to the first 3 hours are shown on each link, $b_j(u, v) = 0$ for all $j = 4, \dots, 23$.

*Input (corresponding to the network map in Figure 1):*

```
1
6 7 55 2
0 1 2 3 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 2 7 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 3 3 2 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 4 4 2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 3 6 1 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 4 5 3 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 5 2 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

*Output:*

```
74
```

**Requirements.** Your algorithm should require time $O(n^3)$ (with reasonable hidden constants).

**Notes.** A reasonable implementation should not require more than 3 seconds for each input file.