



旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

旅游巴士

CSP 2023 Junior Task4

河南省实验中学信息技术组

2023 年 12 月 31 日



注意事项

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

- 输入输出文件: `bus.in/out`
- 时间限制: `1s`
- 内存限制: `512MiB`
- 编译命令: `g++ -O2 -std=c++14 bus.cpp -o bus`



题目描述

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

小 Z 打算在国庆假期期间搭乘旅游巴士去一处他向往已久的景点旅游。

旅游景点的地图共有 n 处地点，在这些地点之间连有 m 条道路。其中 1 号地点为景区入口， n 号地点为景区出口。我们把一天当中景区开门营业的时间记为 0 时刻，则从 0 时刻起，每间隔 k 单位时间便有一辆旅游巴士到达景区入口，同时有一辆旅游巴士从景区出口驶离景区。

所有道路均只能**单向通行**。对于每条道路，游客步行通过的用时均为恰好 1 单位时间。

小 Z 希望乘坐旅游巴士到达景区入口，并沿着自己选择的任意路径走到景区出口，再乘坐旅游巴士离开，这意味着他到达和离开景区的时间都必须是 k 的**非负整数倍**。由于节假日客流众多，**小 Z 在旅游巴士离开景区前只想一直沿着景区道路移动，而不想在任何地点（包括景区入口和出口）或者道路上停留。**

出发前，小 Z 忽然得知：景区采取了限制客流的方法，对于每条道路均设置了一个“开放时间” a_i ，游客只有**不早于 a_i 时刻**才能通过这条道路。

请帮助小 Z 设计一个旅游方案，使得他乘坐旅游巴士离开景区的时间尽量地早。



输入输出格式

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

【输入格式】

输入的第一行包含 3 个正整数 n, m, k ，表示旅游景点的地点数、道路数，以及旅游巴士的发车间隔。

输入的接下来 m 行，每行包含 3 个非负整数 u_i, v_i, a_i ，表示第 i 条道路从地点 u_i 出发，到达地点 v_i ，道路的“开放时间”为 a_i 。

【输出格式】

输出一行，仅包含一个整数，表示小 Z 最早乘坐旅游巴士离开景区的时刻。如果不存在符合要求的旅游方案，输出 **-1**。



输入输出样例

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

【样例输入】

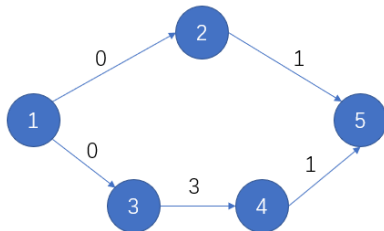
```
5 5 3
1 2 0
2 5 1
1 3 0
3 4 3
4 5 1
```

【样例输出】

```
6
```

【样例解释】

小 Z 可以在 3 时刻到达景区入口，沿 $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$ 的顺序走到景区出口，并在 6 时刻离开。





数据范围

旅游巴士

河南省实验中学
信息技术组

对于所有测试数据有： $2 \leq n \leq 10^4$ ， $1 \leq m \leq 2 \times 10^4$ ， $1 \leq k \leq 100$ ， $1 \leq u_i, v_i \leq n$ ， $0 \leq a_i \leq 10^6$ 。

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

测试点编号	$n \leq$	$m \leq$	$k \leq$	特殊性质
1 ~ 2	10	15	100	$a_i = 0$
3 ~ 5				无
6 ~ 7	10^4	2×10^4	1	$a_i = 0$
8 ~ 10				无
11 ~ 13			100	$a_i = 0$
14 ~ 15				$u_i < v_i$
16 ~ 20				无



子任务 1

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

- 对于测试点 6 ~ 7, 有 $k = 1, a_i = 0$ 。
- 此时, 每条边都没有开始时间限制, 并且可以任意时间出发任意时间到达, 故而 0 时刻从 1 号地点出发走最短路到达 n 号地点的最短路即为答案。
- 时间复杂度: $O(N + M)$ 。
- 预计得分: 10 分。



子任务 2

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

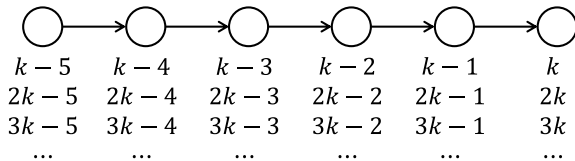
子任务 1

子任务 2

子任务 3

致谢

- 对于测试点 1 ~ 2, 6 ~ 7, 11 ~ 13, 有 $a_i = 0$ 。
- 每条边都没有开始时间限制, 那么显然 0 时刻从入口 1 出发, 到达出口的时间可能是任意时间, 但是要求到达出口 n 的时间只能是 $k, 2k, 3k, \dots$ 。
- 同理到达 n 的入点的时间可能是任意时间, 但是合理的时间应是 $k-1, 2k-1, 3k-1, \dots$, 它们的共同点是对 k 取余的余数均为 $k-1$ 。
- 每个顶点到达的时间都不确定, 但是若想在 $k, 2k, 3k, \dots$ 到达出口, 那么最短路径上的点的到达时间对 k 取余的值一定是有要求的。





子任务 2

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

- 每个顶点到达的时间都不确定，而且需要的到达时间对 k 取余的结果可能是 $0 \sim k-1$ ，故而对每个点按照模 k 构建构建一个 k 层分层图，然后在分层图上求最短路。
- 定义 $d[x][i]$ 表示到达顶点 x 满足时间模 k 等于 i 的最短时间，那么对于道路 (x, y) ，如果有 $d[y][(i+1)\%k] > d[x][i] + 1$ ，那么

$$d[y][(i+1)\%k] = d[x][i] + 1$$

显然 $d[1][0] = 0$ ，答案为 $d[n][0]$ 。

- 时间复杂度： $O(NK)$ 。
- 预计得分：35 分。



子任务 2

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

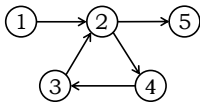
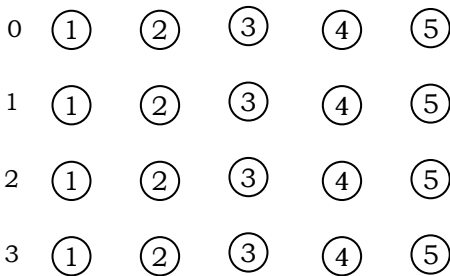
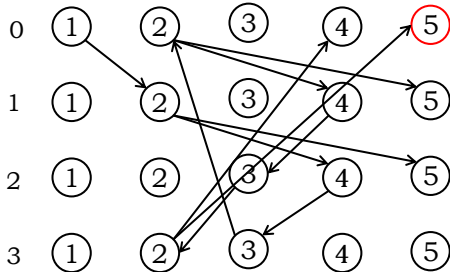


图: $k = 4$

对 k 取余



对 k 取余





子任务 2

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

```
1 // d[x][i] 表示到 x 点满足时间模 k=i 的最短时间
2 int d[N][110];
3 memset(d, 0x3f, sizeof(d));
4 d[1][0] = 0;
5 queue<pair<int, int> > q; q.push({1, 0});
6 while(!q.empty())
7 {
8     // mx 表示最短路 d[x][mx] 对 k 取余的余数
9     int x = q.front().first, mx = q.front().second; q.pop();
10    for(int i = head[x]; i; i = nxt[i])
11    {
12        int y = ver[i], my = (mx + 1) % k;
13        if(d[y][my] > d[x][mx] + 1) q.push({y, my}), d[y][my] = d[x][mx] + 1;
14    }
15 }
16 if(d[n][0] != INF) printf("%d\n", d[n][0]);
17 else puts("-1");
```



子任务 3

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

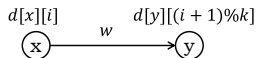
子任务 2

子任务 3

致谢

- 对于一般情况, $a_i \geq 0$ 。
- 有若干条道路有开放时间限制, 也即不能早于某个时间经过。
- 考虑边 (x, y, w) , 假设到达 x 的时间 $d[x][i] < w (0 \leq i \leq k-1)$, 那么为了保证能走 (x, y) 那么就至少需要等待 $w - d[x][i]$, 但是无法停留, 故而有两种选择:

① 晚点进入景区;



② 经过其他点再次到达 x 。

但是不管怎样“拖延”, 拖延的时间必须是 k 的整数倍, 因为要求走到 y 的时间是 $d[y][(i+1)\%k]$ 。

- 故而, 在进行 BFS 扩展时, 如果到达 x 的时间早于道路开放时间 w , 则需要将从 x 出发的时间拖延到大于等于 w 且拖延的时间是 k 的最小整数倍。
- 时间复杂度: $O(NK)$ 。
- 预计得分: 100 分。



子任务 3

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

```
1 // d[x][i] 表示到 x 点满足时间模 k=i 的最短时间
2 int d[N][110];
3 memset(d, 0x3f, sizeof(d));
4 d[1][0] = 0;
5 queue<pair<int, int> > q; q.push({1, 0});
6 while(!q.empty())
7 {
8     // mx 表示最短路 d[x][mx] 对 k 取余的余数
9     int x = q.front().first, mx = q.front().second; q.pop();
10    for(int i = head[x]; i; i = nxt[i])
11    {
12        int y = ver[i], w = a[i], my = (mx + 1) % k;
13        int t = d[x][mx];
14        if(t < w) // 如果走到 x 时 x->y 的开启时间为 w
15        {
16            if((w - t) % k == 0) t = w;
17            else t += ((w - t) / k + 1) * k;
18        }
19        if(d[y][my] > t + 1) q.push({y, my}), d[y][my] = t + 1;
20    }
21 }
22 if(d[n][0] != INF) printf("%d\n", d[n][0]);
23 else puts("-1");
```



致谢

旅游巴士

河南省实验中学
信息技术组

注意事项

题目描述

输入输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

致谢

谢谢观看！