翻修道路

有 n 个城市，用 m 条单向道路连接。

第 i 条道路的长度是 a_i，如果对其进行翻修，则长度变为 b_i。

1 号城市是首都，有 k 个省会城市。

你希望首都到省会城市的最短距离的最大值尽可能小，但是你不能翻修太多道路。
对于所有 $x \in [0, m]$ 求出翻修 x 条道路之后最小的最大值。

输入格式

第一行三个正整数 n, m, k。

第二行 k 个正整数，第 i 个 为 p_i，表示所有省会城市。

接下来 m 行，第 i 行四个正整数 x_i, y_i, a_i, b_i 表示 x_i 到 y_i 有一条单向道路，翻修前后长度为 a_i, b_i。

输出格式

一行 $m + 1$ 个正整数，分别表示 $x = 0, 1 \cdots m$ 时的答案。

样例输入

```
3 3 2
2 3
1 2 12 5
1 3 9 8
2 3 5 2
```

样例输出

```
12 9 7 7
```

样例解释

不翻修道路，1 到 2 的最短距离为 12，1 到 3 为 9，答案为 12。

翻修第一条道路，1 到 2 的最短距离变为 5，答案为 9。

翻修第一条和第三条道路，1 到 3 的最短距离变为 7，答案为 7。

数据范围

保证从 1 号点出发能到达所有点，$k < n$，$p_i \in [2, n]$ 且互不相同，$x_i, y_i \in [1, n]$，
$1 \le b_i \le a_i \le 10^5$。

不保证无重边自环。

$n, m \le 100, k \le 8$。

Subtask 1（20pts）：$n, m \le 10$。

Subtask 2（20pts）：$k = 1$。

Subtask 3（20pts）：$k = 2$。

Subtask 4（40pts）：无特殊限制。

时间限制：1s。

空间限制：256MB。

题解

对于 $n, m \le 10$ 的数据，枚举每条道路是否被翻修，然后用 Dijkstra 算法求出以 1 为源点的最短路，时间复杂度 $O(2^m \log m)$。

对于 $k = 1$ 的数据，考虑分层图最短路，将每个点 i 拆成 $m+1$ 个点 $(i, j), j \in [0, m]$，表示走到点 i 的时候已经经过了 j 条被翻修的道路。(x_i, j) 向 (y_i, j) 连长度为 a_i 的边，(x_i, j) 向 $(y_i, j+1)$ 连长度为 b_i 的边。求以 $(1, 0)$ 为源点的最短路，$x = j$ 的答案为 $(1, 0)$ 到 (p_1, j) 的最短路。$O(m^2 \log m)$。

对于 $k = 2$ 的数据，一定存在一条 1 到 p_1 的最短路和一条 1 到 p_2 的最短路，满足存在一个点 c，两条路径都经过了 c 且之后不存在公共点，并且在经过 c 之前的路径相同。证明：假设不相同，那么这两条 1 到 c 的路径，长度一定相同（否则就不是最短路了），将其中一条替换为另一条即可。以 $1, p_1, p_2$ 为源点分别求分层图最短路（注意以 p_1, p_2 为源点时要将所有边反向），然后枚举 c，再枚举 $1 \to c, c \to p, c \to p$上被翻修的道路数量即可。$O(nm^3)$，可以做到更优但没必要。

对 $k = 2$ 的情况一般化，使用一种类似求最小斯坦纳树的 dp 求解。依次加入 1 到 p_i 的最短路，通过对 1 到 c 的路径调整，可以使这些最短路形成一棵以 1 为根的树。

在这棵树上从叶子到根 dp。$f_{S,i,j}$ 表示当前子树的根是 i，子树内翻修了 j 条道路，子树内包含的省会城市集合是 S，i 到 S 中点的最短距离的最大值的最小值是多少。初始化 $f_{\{p_i\}, p_i, j} = 0$。

按照 S 从小到大的顺序 dp（保证对于 $T \subsetneq S$，T 的 dp 值在 S 之前求出）。第一种情况，集合 S 改变，也就是把两个根相同的子树合并成同一个，$f_{S,i,j+l} \leftarrow \min(f_{S,i,j+l}, \max(f_{T,i,j}, f_{S-T,i,l}))$，枚举 S 的子集即可。第二种情况，集合 S 不变，根从 i 变为 i'，增加一条 i' 到 i 的边，权值为 (a, b)，$f_{S,i',j} \leftarrow \min(f_{S,i',j}, \min(f_{S,i,j} + a, f_{S,i,j-1} + b))$,发现这个转移类似最短路的三角不等式，可以通过一次分层图最短路更新 dp 值。$x = j$ 的答案即为 $f_{\{p_1, p_2 \cdots p_k\}, 1, j}$。

发现复杂度瓶颈是第一种情况，需要求形如 $c_{j+l} = \min(\max(a_l + b_j))$，直接计算是 $O(m^2)$ 的，注意到道路翻修后长度一定不增，所以答案关于翻修道路条数单调不增，a, b 也单调不增，

不妨设 $b_j \ge a_l$，那么可以取 l 为满足 $a_l \le b_j$ 的最小值，这样可以最小化翻修的道路数量，然后再对 c 取一遍前缀 $\min$，就可以 $O(m)$ 计算。$O(2^k m^2 \log m + 3^k nm)$。