

# Solution

---

## Problem 1

---

### Task1

搜索出所有的方案

### Task2

1. 经过猜测和推理，可以发现每次找到剩下的点中权值最大的点就行,复杂度 $O(n^2 + m)$
2. 其他的一些未知做法

### Task3

由2.1可以轻松得到正解：没必要每次都找一遍权值最大的，只需要对所有点按点权排序后依次删除。复杂度 $O(n \log n + m)$ 。

其实我们还可以发现没有必要进行真正的排序，对于一条边 $(a, b)$ ，它的贡献一定是 $\min(a, b)$

## Problem 2

---

### Task1 对于30%的数据 所有菜市位置和小T所在的位置在一条水平直线上

考虑有部分菜市在小T右边，部分菜市在小T左边。只有两种遍历情况：向左再向右、向右再向左 而在每个点处的、可行的最大体格是确定的，因此直接累加即可。最终答案在两种情况中选取一个路径更短的或最短路径相同情况下体格和最大的。

### Task2 对于s=0且n,m<=50的数据

当 $s = 0$ 时，问题转变为从一个点出发遍历 $p$ 个点的最短路问题。最朴素的想法应该是搜索，每次搜索下一步去哪个点。但搜索的问题在于它考虑了经过每个点的顺序。实际上经过顺序不会对答案产生影响、经过了哪些点才会有影响。因此考虑用状压表示哪些点没去过，然后寻找一个没去过的点更新。定义状态 $dis[x][y][s]$ 表示还没去过 $s$ 集合中的点，目前在点 $(x, y)$ 的最短距离。可使用bfs更新，复杂度 $O(n \times m \times 2^p)$ ，可解决 $n, m \leq 50$ 的情况。

### Task3 对于s=0的数据

很容易可以发现， $dis[x][y][s]$ 的三维中， $s$ 在很多点是不会被更新的，因此这些点处出现了状态冗余。因此只在 $p$ 个点处定义 $dis$ ，即状态改为 $dis[i][s]$ ，表示在第 $i$ 个点处的最短路。转移时需要利用两个点间的最短路。最短路可通过 $O(p)$ 次bfs预处理得到。

复杂度 $O(p \times n \times m + 2^p \times p^2)$

### Task4 对于p=1的数据

只有一个目标点，因此只用考虑膨胀问题。

在每一个点都会重新膨胀，因此在每个点处的最大膨胀值其实是固定的。

在bfs的时候直接将每个点的膨胀值加入答案即可。

复杂度 $O(p \times n \times m + s^2 \times n \times m)$

## Task5 对于100%的数据

将2.3中的bfs部分与2.4结合，即可得到正解。

复杂度 $O(p \times n \times m + s^2 \times n \times m + 2^p \times p^2)$

## Problem 3

---

### Task1

对于每次查询，暴力搜索到  $n$  的路径。

### Task2

注意到每次转移的状态只和当前点  $u$  以及从  $s$  到当前点经过的所有边的  $gcd$  的值  $g$ 。

所以很容易可以想到一个  $dp$ 。

定义  $dis[u][g]$  为从起点到  $u$  点，当前经过的所有边的  $gcd$  为  $g$  的最短路。

转移时枚举  $u$  点的出边  $(u, v, w)$ ，定义  $g' = gcd(g, w)$ 。则转移方程如下：

$$dis[v][g'] = \min(dis[u][g] + \frac{w}{g'})$$

由于每个点的状态数只与边的权值的因数有关，故总状态数为  $m \times \sqrt{m}$ 。dp 的转移借助dijkstra 进行，一次查询的复杂度为  $m \times \sqrt{m} \times \log n$ ， $Q$  次查询的复杂度为  $Q \times m \times \sqrt{m} \times \log n$ （最后一个  $\log$  为 dijkstra 的复杂度），预计得分 60

### Task3

很容易注意到，终点是固定的，所以可以从这里下手。

但是有一点不是很方便，就是边权的定义。但边权只与起点到这条边的  $gcd$  有关，因此可以枚举这个  $gcd$  的值。

定义  $dis[u][g]$  为从  $u$  点到终点，从起点到  $u$  点的所有路径的  $gcd$  为  $g$ 。

转移时枚举  $v$  点的出边  $(u, v, w)$ ，枚举  $g'$  的倍数  $g$ 。注意到该转移方程只可在  $g' | w$  的时候进行转移

$$dis[u][g] = \min(dis[v][g'] + \frac{w}{g'})$$

虽然需要枚举  $g$ ，但该算法可以  $O(1)$  地回答每个询问

该算法复杂度为  $O(m \times V \times \log V + Q)$ （最后一个  $\log$  为 dijkstra 的复杂度），足以通过此题