

# 同余最短路

当出现形如“给定  $n$  个整数，求这  $n$  个整数能拼凑出多少的其他整数（ $n$  个整数可以重复取）”，以及“给定  $n$  个整数，求这  $n$  个整数不能拼凑出的最小（最大）的整数”的问题时可以使用同余最短路的方法。

同余最短路利用同余来构造一些状态，可以达到优化空间复杂度的目的。

类比 [差分约束](#) [../diff-constraints/] 方法，利用同余构造的这些状态可以看作单源最短路中的点。同余最短路的状态转移通常是这样的  $f(i + y) = f(i) + y$ ，类似单源最短路中  $f(v) = f(u) + \text{edge}(u, v)$ 。

## 例题

 **P3403 跳楼机** [<https://www.luogu.com.cn/problem/P3403>]

题目大意：给定  $x, y, z, h$ ，对于  $k \in [1, h]$ ，有多少个  $k$  能够满足  $ax + by + cz = k$ 。

不妨假设  $x < y < z$ 。

令  $d_i$  为只通过 **操作 2** 和 **操作 3** 能够达到的最低楼层  $p$ ，并且满足  $p \bmod x = i$ 。

可以得到两个状态：

- $i \xrightarrow{y} (i + y) \bmod x$
- $i \xrightarrow{z} (i + z) \bmod x$

注意通常选取一组  $a_i$  中最小的那个数对它取模，也就是此处的  $x$ ，这样可以尽量减小空间复杂度（剩余系最小）。

那么实际上相当于执行了最短路中的建边操作：

```
add(i, (i+y) % x, y)
```

```
add(i, (i+z) % x, z)
```

接下来只要求出  $d_0, d_1, d_2, \dots, d_{x-1}$ ，只需要跑一次最短路就可求出相应的  $d_i$ 。答案即为：

$$\sum_{i=0}^{x-1} \frac{h - d_i}{x} + 1$$

加一是由于当前所在楼层也算一次。

#### 参考实现

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  const int maxn = 100010;
6  const int INF = 0x3f3f3f3f;
7
8  ll h, x, y, z;
9  ll head[maxn << 1], tot;
10 ll dis[maxn], vis[maxn];
11 queue<int> q;
12
13 struct edge {
14     ll to, next, w;
15 } e[maxn << 1];
16
17 void add(ll u, ll v, ll w) {
18     e[++tot] = edge{v, head[u], w};
19     head[u] = tot;
20 }
21
22 void spfa() {
23     dis[1] = 1;
24     vis[1] = 1;
25     q.push(1);
26     while (!q.empty()) {
27         int u = q.front();
28         q.pop();
29         vis[u] = 0;
30         for (int i = head[u]; i; i = e[i].next) {
31             int v = e[i].to, w = e[i].w;
```



```
32         if (dis[v] > dis[u] + w) {
33             dis[v] = dis[u] + w;
34             if (!vis[v]) {
35                 q.push(v);
36                 vis[v] = 1;
37             }
38         }
39     }
40 }
41 }
42
43 int main() {
44     memset(dis, INF, sizeof(dis));
45     scanf("%lld", &h);
46     scanf("%lld %lld %lld", &x, &y, &z);
47     if (x == 1 || y == 1 || z == 1) {
48         printf("%d\n", h);
49         return 0;
50     }
51     for (int i = 0; i < x; i++) {
52         add(i, (i + z) % x, z);
53         add(i, (i + y) % x, y);
54     }
55     spfa();
56     ll ans = 0;
57     for (int i = 0; i < x; i++) {
58         if (h >= dis[i]) ans += (h - dis[i]) / x + 1;
59     }
60     printf("%lld\n", ans);
61     return 0;
62 }
```

## 习题

洛谷 P3403 跳楼机 [<https://www.luogu.com.cn/problem/P3403>]

洛谷 P2662 牛场围栏 [<https://www.luogu.com.cn/problem/P2662>]

[国家集训队]墨墨的等式 [<https://www.luogu.com.cn/problem/P2371>]



[NOIP2018] 货币系统 [<https://loj.ac/problem/2951>]