迷宫探险

正着做最短路似乎有点困难，于是我们可以倒着从终点开始跑最短路。

令 dis_u 表示 u 到终点的最短路，那么我们仍然可以用朴素 dij 算法来转移，更重要的是按照 dij 从小到大转移的顺序，每次转移到一个点的时候这个点就必须利用剩余的荆棘覆盖转移来的边，所以当无法覆盖时就可以转移。

划分区间

考虑简单一点的问题，即只需要最小化所有线段的最大值。

当然可以二分，但是我们提供另一种解决方案，令 $v(l,r,0/1,0/1)$ 表示线段树上区间 $[l,r]$ 内部已经全部覆盖，覆盖 l 的区间左端点是否为 l，覆盖 r 的区间右端点是否为 r。

合并就是 $v(l,r,x,y)=\min(\max(v(l,mid,x,z),v(mid+1,r,z,y)))$，可以直接线段树维护。线段树叶子节点的初值根据定义算即可。

而对于最小值我们可以枚举最小值 V，那么相当于权值小于 V 的区间不能使用，放在线段树上就是把这些区间设为 $+\infty$。

于是每次只需要枚举最小值，在线段树上单点修改后全局查询即可。因为最多只有 $2n-1$ 个线段所以枚举 V 的复杂度可以接受。

时间复杂度 $\mathrm{O}(n\log n)$。

西西弗西

我们可以考虑枚举 s 的后缀作为 A。那么会贡献给所有前缀是 AA 且满足长度大于 $3|A|$ 的后缀一次。

如果我们将所有后缀按照字典序排序，那么可以发现是 AA 前缀的后缀是一个区间，所以我们只要找到了这个区间，问题就变为了二维数点问题，用树状数组可以简单解决。

对于如何找到这个区间，我们可以二分，我们通过二分找到第一个前 $|AA|$ 个字符 $\geq AA$ 的字符串，再找到前 $|AA|$ 个字符 $> AA$ 的字符串，那么中间的左闭右开区间就是我们的目标。

上述操作种的排序，二分等过程，均需要快速查询一对后缀的 lcp，可以用哈希快速计算。时间复杂度 $\mathrm{O}(n \log^2 n)$。

当然如果你会 SA 那么可以把过程优化到 $\mathrm{O}(n \log n)$

二分图匹配

首先一个观察，如果把所有 a_i 从大到小排序，令 $b=\max(l_i+i-1)$，其中 l_i 为 a_i 最高位，那么 s 的最高位要么为 b 要么为 $b+1$。

因为如果我们依次考虑，假设上一个位置最高位填了 x，那么接下来的若干位置，第 i 个位置的最高位不能超过 $x-i+1$。所以下界是 b。

同时如果令 $a_i=2^{b-i+2}$，显然是一组合法解，故上界是 $b+1$。

考虑如何判断是 b 还是 $b+1$，我们还是依次考虑，假设目前考虑到了 i，如果 i 满足 $l_i+i-1=b$，那么这个位置就是一个临界点，换句话说，这个数必须选择 l_i 这一位为 1。

那 a_i 后面的数怎么办呢。其实我们只要找到它最高位下的第一个 1，然后把这个后缀取出来按照大小顺序再插入到后面就行了，于是变成了一个子问题。

当出现了一个位置有 $l_i+i-1>b$ 时，就一定不合法，否则就合法。于是我们会检查最高位了。

上述过程其实已经基本给出了一个构造方法，即对于非临界的位置，可以直接设为 2^{l_i+1}，对于临界位设为 2^{l_i}。

当然这是完全检查成功的情况，还有可能检查失败，但是可以发现这种情况下第一个失败的临界位之前的数是确定的，之后的数又是一个子问题。

于是我们会了全部流程，只需要快速维护这个东西就好了。

首先因为要给所有后缀按大小插入到检测序列里，所以要给所有后缀排序。这是简单的，只需要按位排列，给每个以 1 开头的后缀排序即可。

接下来我们需要快速找到第一个满足 $l_i+i-1=b$ 的数以及支持插入，可以使用平衡树，也可以用把所有后缀先插入后用线段树维护。

因为 1 的数量是可以接受的，所以上面维护的复杂度都是 $\mathrm{O}(n\log n)$。