Solutions
2019 ECNU XCPC July Selection #2[1]

NULL[2]

ECNU

July 2019

[1] powered by XeLaTeX
[2] NULL's wiki

Summary

总览

- 考察的知识点
  - A easy 数据结构 DP(经典模型)
  - B easy BFS
  - C easy DP($0/1$ 背包)
  - D hard DFS DP(状态压缩)
  - E mid 最短路
  - F easy 模拟 高精度
- 难度的定义是针对本次比赛的题目, 这些题目除了 D , 在区域赛都是签到级别
- 对前几次专训的综合考察, 难度总体较低

Summary

F. 格雷码

- 构造的方法已经给出了：
  - 顺序书写 N 位二进制反射格雷码，然后在前面加上 0；
  - 逆序书写 N 位二进制反射格雷码，然后在前面加上 1。
- 所以我们只需要知道他位于 N 位格雷码前半段还是后半段就知道第一位
- 我们只需要知道他位于 $N-1$ 位格雷码前半段还是后半段就知道第二位
- 递归下去就是了
- 需要高精度

F. 格雷码

- 高精度部分直接用 Python 会很简洁

```
1  n,k=map(int,input().split())
2  t=1<<n; k-=1; ans=0
3  while t:
4      if k>=t:
5          ans+=t
6          k=t*2-1-k
7      t>>=1
8  print(ans)
```

Summary

B. 烤乐滋逃亡

- 所有树在的位置为 0 , 我们令树在的位置为起点, 跑最短路
- 每个格子的代价就是所有起点的最短路代价的 min
- 由于路径代价都是 1 , 我们直接 BFS 就好了
- 时间复杂度 $O(nm)$

Summary

A. 阿卡的萝卜

- 区间加, 可以离线的话, 有一个古老的套路
  - 在开始位置和结束位置分别打上标记
  - 从头到尾扫一遍, 就能得到每一个位置的最终答案
  - 时间复杂度 $O(n)$
- 知道了所有位置的值以后, 需要做的就是连续区间最大和
- 经典的 dp 模型

Summary

C. 阿瓦的礼物

- 每个礼物要么没有炸弹, 要么有 V/K 个炸弹
- 可以转换成 $0/1$ 背包问题
- $f[i]$ 表示有 i 个炸弹的方案数
- 转移的时候分两种情况:
  - 假设这个礼物有 $x = V/k$ 个炸弹, $f[i+x] += f[i]$
  - 假设这个礼物没有炸弹, $f[i] += f[i]$

Summary

E. 最短路

- 如果把起点和终点也看成强力充电站的话, 问题可以做一个转换
- 电动车能不能从起点到终点, 途中的点只剩下强力充电站, 边只能是 $\leq k$ 的才能跑
- 以每一个点强力充电站为起点跑一遍最短路, 重新建图
- 再跑起点到终点的最短路
- 多源最短路, floyd 的复杂度不被接受

Summary

D. Bits

- 你们好像几乎不补题, 那再来一次。
- $S_1..S_{n-m}$ 和 $S_{m+1}..S_n$ 匹配, 我们可以很显然的得出, $S_i = S_{i+m}$
- 也就是说, 我们把原串按照 m 分块以后, 分别对应每一块的第一位、第二位都要相等
- 分两种情况讨论:
  - $m \leq \sqrt{n}$
  - 这样的情况, 每一块都很小, 我们考虑枚举块的最终状态, $f[i][j][k]$ 表示, 到第 i 块, 每一块的最终状态为 j, 当前是否处于反转状态 $k = 0/1$。
  - 这样就很好转移了, 因为后面的反转状态和前面的无关, 从后往前推 dp 即可。

D. Bits

- 另一种情况:
  - $m > \sqrt{n}$
  - 这个每一块都很大, 但是块不多
  - 我们可以发现, 对于反转 $S_1..S_{k*m}(k = const)$ 的操作, 对于每一个 k 最多执行一次, 因为执行多了没有意义, 而且执行的先后任意
  - 这样, 我们就可以枚举, 每一个 k 时候执行, 然后再暴力计算还需要的第一类操作次数即可

END

Solutions
2019 ECNU
XCPC July
Selection #2

NULL

Thanks!