**D13ASTR**

* Chuẩn bị trước mảng Next[i][j] với ý nghĩa là độ dài trùng xâu tiền tố khi thêm j trong lúc đã trùng với xâu tiền tố độ dài i.
* Sub 1 + 2 : ta quy hoạch động với f[i][j] là số cách khi đến vị trí i và trùng với tiền tố xâu S độ dài j

f[i+1][next[d][k]] = f[i + 1][Next[d][k]] + f[i][d]

* Kết quả của bài toán sẽ là tổng các f[n][i] ( 0 <= i < n)
* Sub 3 : ta sẽ cải thiện bằng kĩ thuật nhân ma trận

Tính ma trận A:

if (Next[j][0] == i) A.v[i][j] ++;

if (Next[j][1] == i) A.v[i][j] ++;

Ta sẽ tính ma trận A nhân lại n lần kết quả bài toán là

kq = (kq + ans.v[i][0]) % MOD;

**XYZ**

* Ta sẽ qui các cột thành bitmask khi đó bài toán trở về như bài d13ASTR (nhưng ở đây có 2 hình nên thêm chiều trùng tiền tố xâu 2 nữa)
* Ta sẽ phải tính mảng Next của cả 2 hình 1 và 2 f[i][j][k] là số cách đến vị trí i mà trùng với tiền tố hình thứ nhất j cột và hình thứ 2 k cột

F[i + 1][nx[1][j][t]][nx[2][k][t] += f[i][j][k]

(0 <= i < n , 0 <= j , k <= l , 0 <= t < 32)

Kết quả bài toán sẽ là :

ans += ( f[n][l][i] + f[n][i][l]) (0 <= i < l)