s	t	
a	b	0
b	b	1
ab	b	1+0
bb	b	1+1
ab	ab	1
bb	ab	0
aabcb	ab	4
aabb	ab	4

Uint64 否则溢出

```
for(int i=[redacted]; i<n; i++)
{   int sum=0;
    for(int j=i; j<=m; j++)
    {
        sum += dp[j][i];
        dp[j+1][i+1] = sum
    }
}
```

减少一层循环
同时只计算矩阵下半部分三角

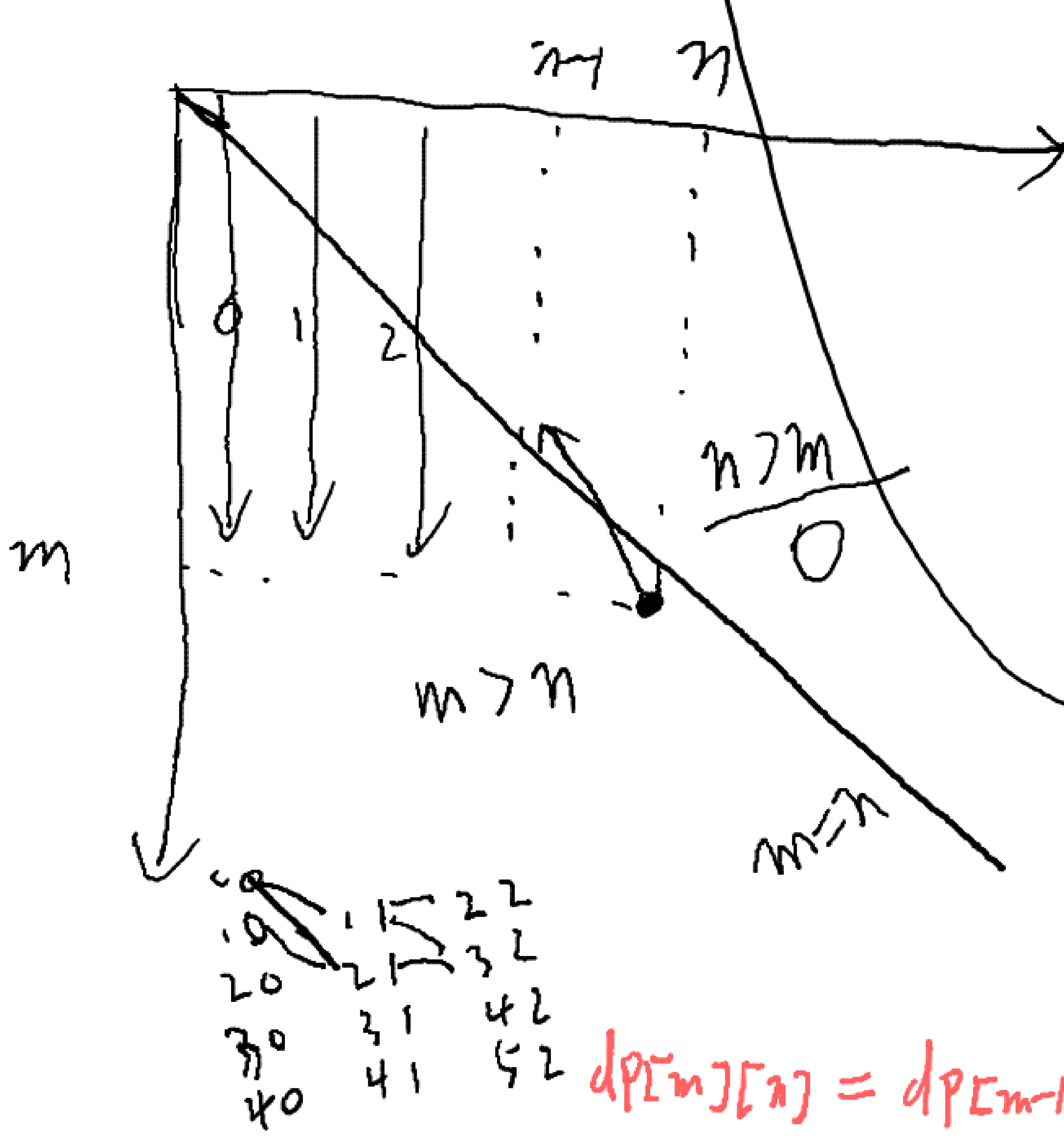
dp[m,n] 表示 S[0:m] 中以 S[m-1] 为最后一个字符能够形成字符串 t[0:t] 的数量.

则最终节果 res[m][n] = dp[m,n] + dp[m-1,n] + ······ ~~dp[0,n]~~ dp[n,n]

dp[m,n] = 0 if m<n 或 S[m] != t[n]

S[m] == t[n]

dp[m,n] = dp[m-1,n-1] + dp[m-2,n-1] + ······ ~~dp[0,n-1]~~ dp[n-1,n-1]



dp[m][n] = dp[m-1][n-1] + dp[m-1][n]

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<=i; j++)
    {
        ~~for dp[j,i]=~~
    }
}
```

三层循环 1140ms, 可以减少

另一种解释

$dp[m][n]$ 代表 $s[0:m]$ 中能形成 $t[0:n]$ 的数量(左闭右闭)

则：

$$dp[m][n] = \begin{cases} \text{1. 使用s的最后一个字符 } s[m] \\ \quad \begin{cases} s[m] \neq t[n] & 0 \\ s[m] == t[n] & dp[m-1][n-1] \end{cases} \\ + \\ \text{2. 不使用s的最后一个字符 } s[m] \\ \quad dp[m-1][n] \end{cases}$$

$dp[m][n]$ $m<n$ 时为0

$m=n$ 时 $s[0:m] == s[0:n]\ ?\ 1 : 0$

$m>n$ 时：

$$dp[m][n] = dp[m-1][n] + ((s[m]==t[n])\ ?\ dp[m-1][n-1] : 0)$$

$$dp[m][n] = \underbrace{dp[m-1][n]}_{\text{为0}} + \underbrace{((s[m]==t[n])\ ?\ dp[m-1][n-1] : 0)}_{\text{展开后即字符按位置进行比较全部相同为1，否则为0}}$$