思路
将每个等级的强化附上一个权重，

$$
\begin{cases}
W_0 = 0\text{（未强化）} \\
W_1 = \text{（+1的次数期望）} \\
W_2 = \text{（+2的次数期望）} \\
W_n = W_{n-1} + W_{n-2}\text{（+n的次数期望）}
\end{cases}
$$

这个是+1、+2强化，后面使用新镜子的策略。
采用母函数的方法，计算了W_1和W_2的值：

$$W_1 = \frac{d}{dx}\left[\frac{p_0 x}{1-(1-p_0)x}\right]_{x=1}$$

$$W_2 = \frac{d}{dx}\left[\frac{p_0 p_1 x^2}{1-p_0(1-p_1)x^2-(1-p_0)x}\right]_{x=1}$$

~~(有卡西欧懒得化简了)~~

然后递推计算出后续的W_n。
我们假设某次采用强化，收益更高。
则有

$$P_i W_{i+1} + (1-P_i) W_{i-1} \geq W_i$$

$$\Rightarrow P_i \geq \frac{W_i - W_{i-1}}{W_{i+1} - W_{i-1}}$$

通过python程序计算出各个等级的临界概率如下：

```python
f=[0,1.8115,5.6594]
# f_1=d[(p0*x)/(1-(1-p0)*x)]/dx at x=1
# f_2=d[(p0*p1*x^2)/(1-(1-p1)*p0*x^2-(1-p0)*x)]/dx at x=1
for i in range(2,20):
    f.append(f[i]+f[i-1])
print([format(x, '.4f') for x in f])
p=[0]*20
for i in range(3,20):
    # f[i]<=(1-p)*f[i-1]+p*f[i+1]=f[i-1]+p*(-f[i-1]+f[i+1])
    # p>=(f[i]-f[i-1])/(-f[i-1]+f[i+1])
    p[i]=(f[i]-f[i-1])/(-f[i-1]+f[i+1])
print([format(x*100, '.2f')+'%' for x in p[3:20]])
```

结果为：

强化等级	临界成功
3	24.25%
4	43.10%
5	36.26%
6	38.93%
7	37.92%
8	38.30%
9	38.16%
10	38.21%
11	38.19%
12	38.20%
13	38.20%
14	38.20%
15	38.20%
16	38.20%
17	38.20%

强化等级	临界成功
18	38.20%
19	38.20%

思路仅供参考，实际情况可能会因为运气波动有所不同。