用dp[i][j][k]表示第i回合，剩余生命为j，剩余青蛙卡牌数为k时对boss造成的最大伤害。初始值dp[0][j][k]=0, 0<j<=M, 0<=k<=K。每一轮先将dp[i+1][j][k]设为-inf。在(i,j,k)状态下有三种选择：

1、回复，会转移到状态(i+1,min(M, j+c[i])-a[i],k)（如果min(M, j+c[i])>a[i]），因此转移方程为dp[i+1][min(M, j+c[i])-a[i]][k]=max(dp[i+1][min(M, j+c[i])-a[i]][k], dp[i][j][k])。

2、攻击，如果b[i]>=H-dp[i][j][k]，那么这一轮攻击可以消灭boss，返回答案i。否则会转移到状态(i+1,j-a[i],k)（如果j>a[i]），因此转移方程为dp[i+1][j-a[i]][k]=max(dp[i+1][j-a[i]][k], dp[i][j][k]+b[i])。

3、使用青蛙卡牌，会转移到状态(i+1, j, k-1)（如果k>0），因此转移方程为dp[i+1][j][k-1]=max(dp[i+1][j][k-1], dp[i][j][k])。

如果始终没有b[i]>=H-dp[i][j][k]的判断，说明无法消灭boss，输出fail。

状态数为O(NMK)，转移时间复杂度为O(1)，因此总时间复杂度为O(NMK)。

使用滚动数组能够使空间复杂度降为O(MK)。