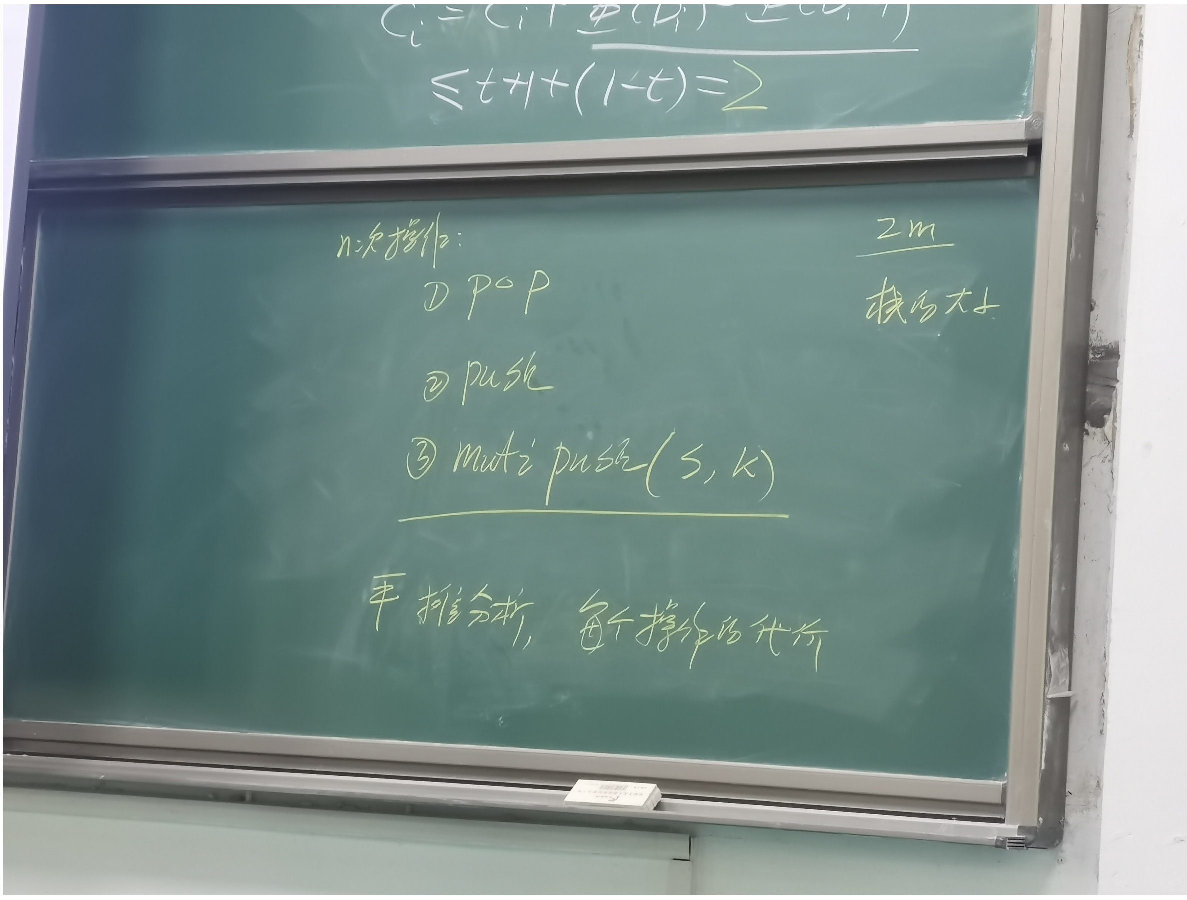
第三次随测

问：假设栈大小为2m，考虑栈上操作pop，push，multipush作摊还分析，每个操作的代价。



为什么做摊还分析?

单个操作的平均代价不好求，最坏代价又不能很好反映该操作实际使用中的代价，如multipush(k个元素)，最坏代价O(2m)，其实际代价一般没那么大，故联合其数据结构上的其他操作做摊还分析，求n个多种操作的总代价，然后摊到单个操作上作为其摊还代价。

方法一：记账法（最简单的）

出现一个元素记下代价2，其中一个代价支付入栈操作，另一个代价支付出栈操作。

若最后栈空，则n个操作最多出现n-1个元素，即开始时一次multipush(n-1个元素)，后续(n-1)次pop将栈清空，记账代价为O(2(n-1))。

若最后栈不空，这时的总代价一定小于等于在栈空基础上增加2m次push操作将栈填满，故摊还总代价为O(2(n-1)+2m)，一个操作的摊还代价O((2(n-1)+2m)/n) = O((n+m)/n)。

方法二：聚合分析（参考119号同学）

一次multipush代价最多O(2m)，最坏情况下假设执行k次pop，n-k次multipush，则有：

$$-k+(n-k)2m=2m$$
$$2m(n-1)=(2m+1)k$$
$$k=\frac{2m(n-1)}{2m+1}$$
$$n-k=\frac{2mn+n-(2mn-2m)}{2m+1}=\frac{2m+n}{2m+1}$$

最坏情况时间：

$$kO(1)+(n-k)O(2m) \qquad = \frac{2m(n-1)}{2m+1}+2m*\frac{2m+n}{2m+1}$$
$$=O(m+n)$$

$$\text{故摊还代价为}O(\frac{m+n}{n})=O(\frac{m}{n})$$

大部分用聚合分析的同学：

（1）假设开始一次multipush，代价O(2m)，后续（n-1）次pop和push，代价O(n)，摊还代价O((2m+n)/n)，这样是不合适的，multipush就一次太少了。

（2）一次multipush(2m个元素)+2m次pop构成一轮，这是片面的，n次操作共n/(2m+1)轮，总代价O(((2m+2m)*n)/(2m+1))，但这样的前提是m<<n，故摊还代价为O(4mn/(2m+1)n)=O(2)

方法三：势函数法（目前还未解决）

假设$\Phi(D_i)=\text{栈}D_i\text{状态下的元素个数}$，故其与书上例题中势函数定义相同。故符合势函数要求

$$\begin{cases}\Phi(D_0)=0\\ \Phi(D_i)\$\geqslant 0\end{cases}$$

其中pop，push的摊还代价仍是0和2。

对于multipush(k个元素)：

$$\hat{C}_i=C_i+\Phi(D_i)-\Phi(D_{i-1})=2C_i\leq 4m$$
$$\text{其中，}C_i=min(k,2m-\Phi(D_{i-1}))\text{，}\Delta=C_i$$
$$\text{摊还代价}\frac{\sum\hat{C}_i}{n}\leq\frac{4mn}{n}=O(m)$$

这样做太大了，说明势函数这样定义不合适。