



Open Topic

用队列实现栈及其正确性的证明



如何用队列实现栈

最容易想到的办法

假设两队列Q1，Q2

入栈

直接全部压入为空的那一个队列

出栈

以空队列作为中转站，将另一队列中除了队尾的元素都出队，并压入空队中。

最后弹出并返回剩余的那个元素

如何用队列实现栈

时间复杂度

入栈 $O(1)$ 出栈 $O(n)$

空间复杂度

入栈 O 出栈 $O(1)$

出栈时只需一个temp在两队列中转移元素

Better Algorithm?

空间复杂度

入栈 $O(1)$ 出栈 $O(1)$ 较为理想无需优化

时间复杂度

入栈 $O(1)$ 出栈 $O(n)$ 对出栈效率进行优化

- 优化出栈效率，也就是要减少队列尾元素之前的元素数量，使其能尽快被弹出队列

如何用队列实现栈（优化）

入栈

直接压入Q1中

维护

设Q1中元素个数 n_1 ，Q2中元素个数 n_2

维护 $n_1^2 \leq n_1 + n_2$ ，否则就将Q1的首元素转移至Q2的尾部

如果Q1为空，就在Q2中循环直至将最后 $\sqrt{n_2}$ 个元素移入Q1。

出栈

在Q1中循环（将首元素移出队列再移至队尾）直至将最后一个元素pop出。

。

Initialize size1=0, size2=0

//algorithm of push(S,x)

add(x,Q1);

size1++;

//algorithm of is-empty(S)

if size1==0&&size2==0

return true;

else return false;

//algorithm of pop(S)

if size1==0&&size2==0

return ERROR;

maintain(S)

do the following *size1-1* times

remove(temp,Q1);

add(temp,Q1);

dequeue(Q1);

//algorithm of maintain(S)

if size1==0

x=[sqrt(size2)] //需要移入Q1的元素个数

do the following size2-x times

remove(temp,Q2);

add(temp,Q2);

do the following x times

remove(temp,Q2);

add(temp,Q1);

size2--;size1++;

while size1>sqrt(size2)

remove(temp,Q1);

add(temp,Q2);

size1--;size2++;

如何用队列实现栈

Better?

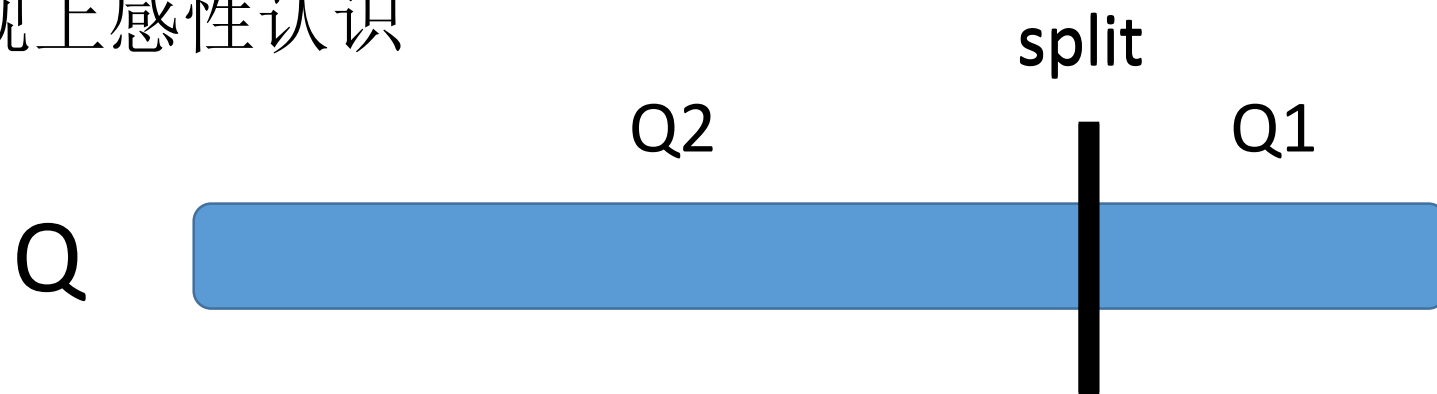
时间复杂度

入栈 $O(1)$ 出栈 $O(\sqrt{n})$

维护 $O(n)$

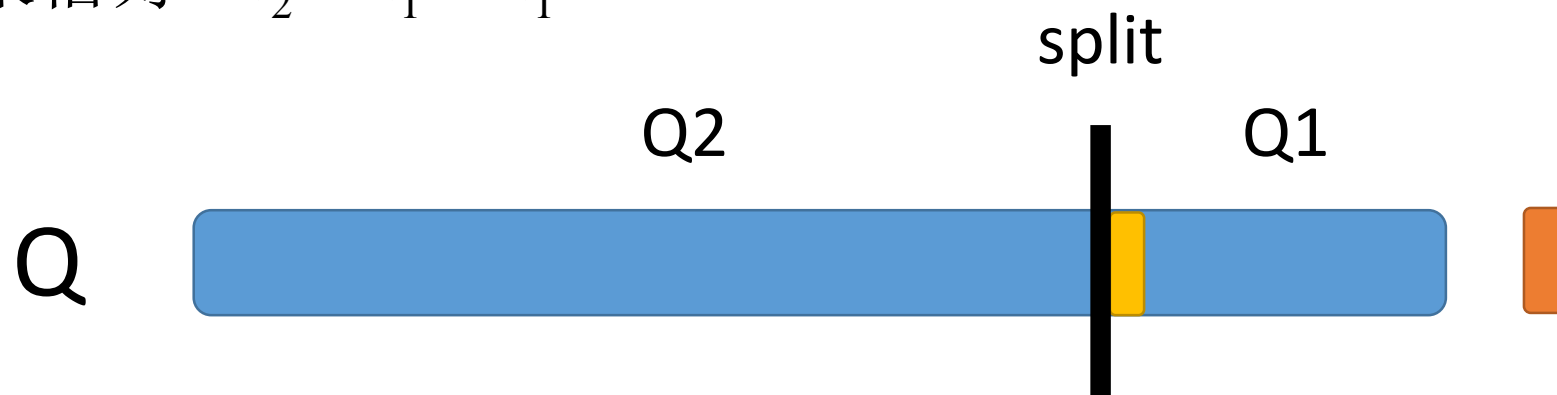
正确性证明

直观上感性认识



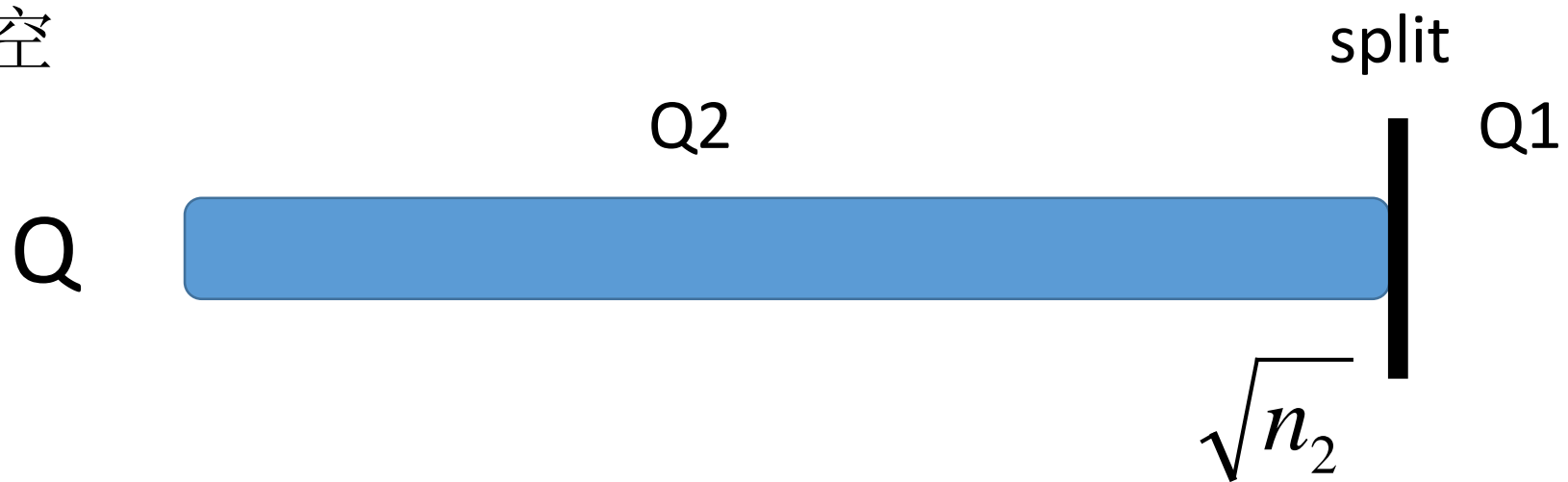
维护 $n_2 + n_1 \geq n_1^2$

如果恰好 $n_2 + n_1 = n_1^2$



正确性证明

若Q1为空



正确性证明

可见，若将Q1、Q2视为一个队列的两部分，每次维护时并不改变这个队列内的元素顺序，只是对分割点进行更改。

所以每次pop出的元素都是在Q1尾部，即整个队列尾的，也就是最后一个进入队列的。符合后进后出原则。



Thanks

