栈与队列

Steap + Queap

04-K

见贤思齐焉，见不贤而自省也

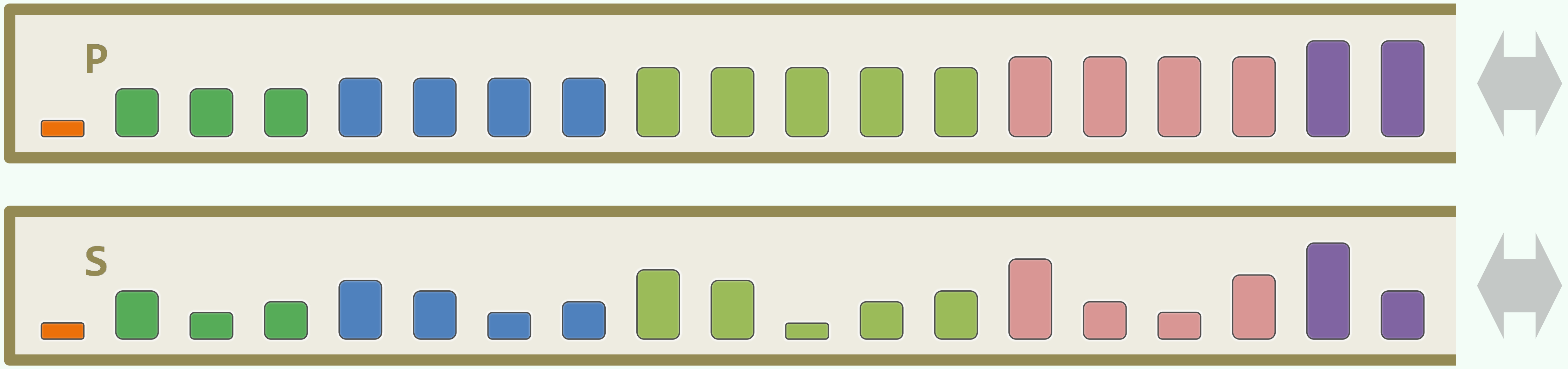
A hero is born among a hundred, a wise man is found among a thousand, but an accomplished one might not be found even among a hundred thousand men.

邓俊辉

deng@tsinghua.edu.cn

Steap = Stack + Heap = push + pop + getMax = S + P

❖ P中每个元素，都是S中对应前缀里的最大者



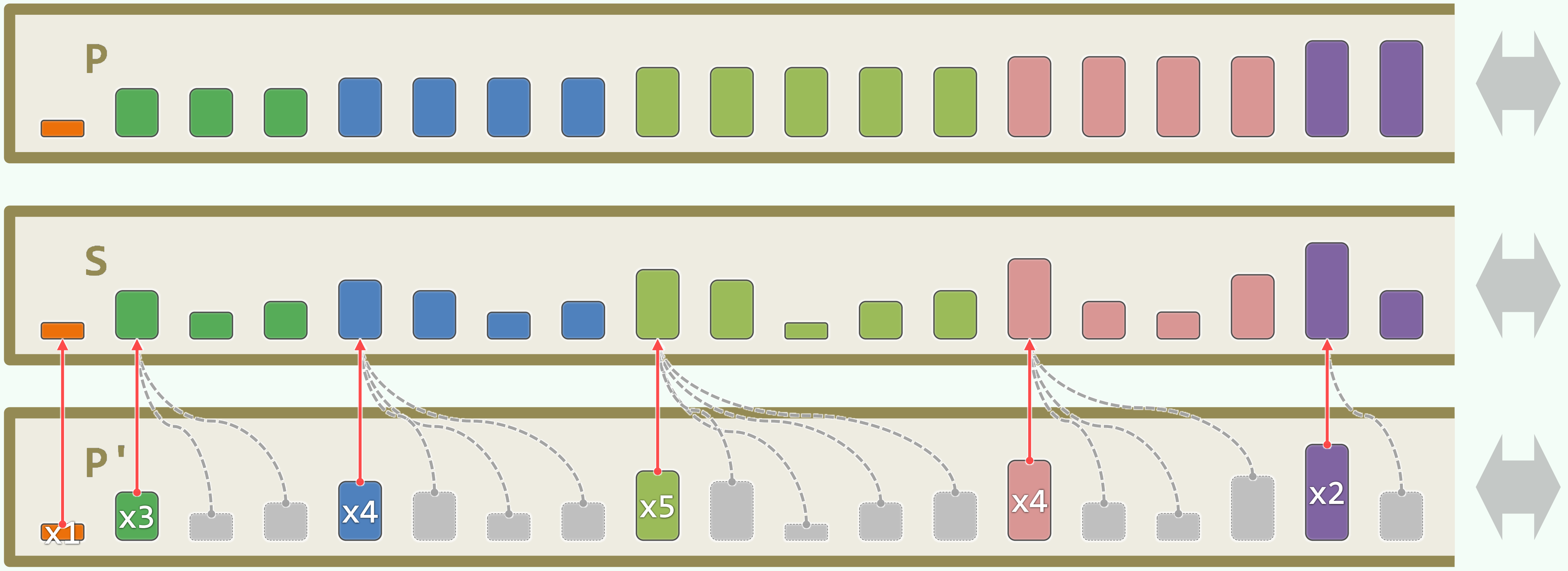
❖ Steap::getMax() { return P.top(); } //$\mathcal{O}(1)$

❖ Steap::pop() { P.pop(); return S.pop(); } //$\mathcal{O}(1)$

❖ Steap::push(e) { P.push(max(e, P.top())); S.push(e); } //$\mathcal{O}(1)$

Steap = Stack + Heap = push + pop + getMax = S + P'

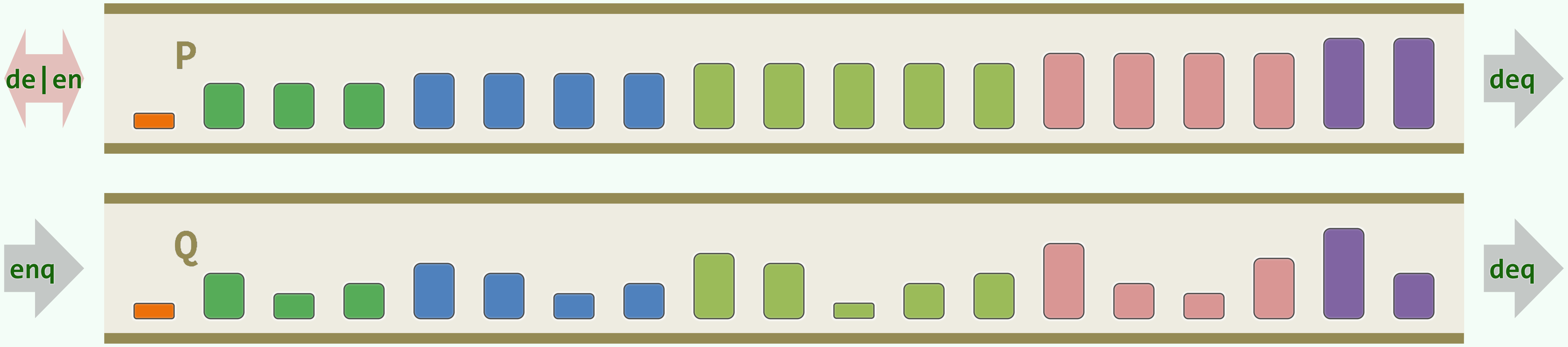
❖ P去重，**升级至**P'：S中的**关键**元素，才会记录在P'中；只需记录对应的**引用**，外加一个**计数器**



❖ S.push()和S.pop()，分别对应于计数器的**增|生**、**减|灭**

Queap = Queue + Heap = enqueue + dequeue + getMax = Q + P

❖ `Queap::dequeue() { P.dequeue(); return Q.dequeue(); } //O(1)`

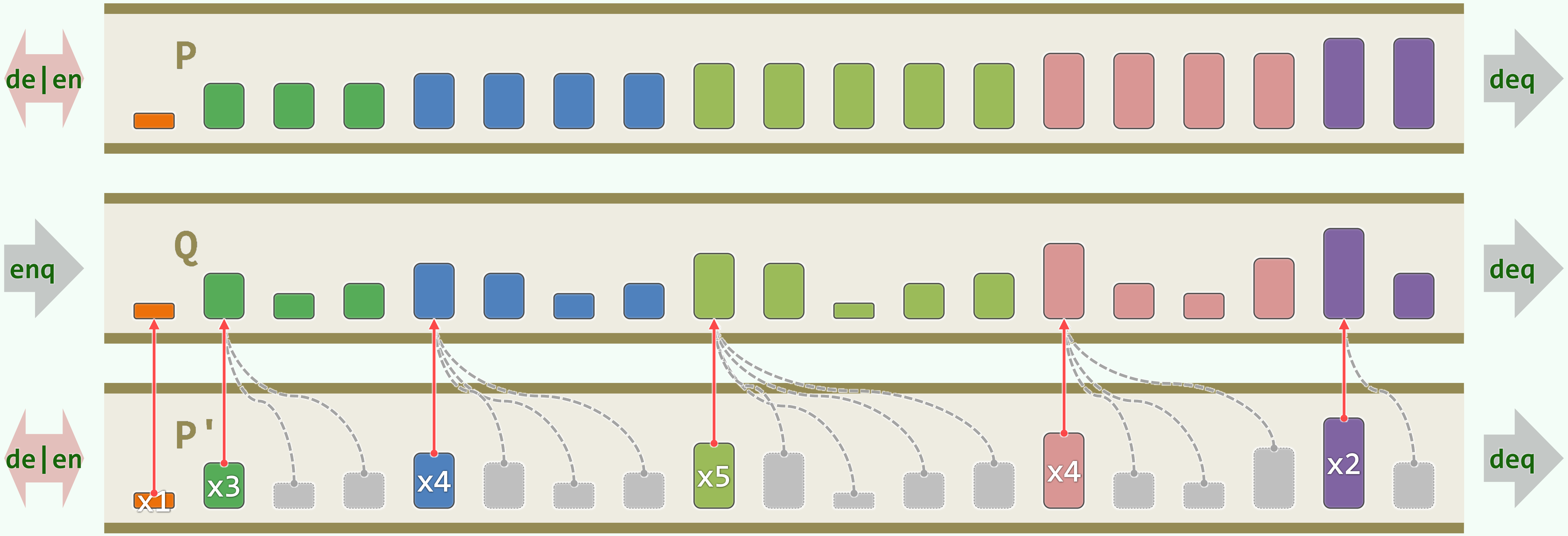


❖ Queap::enqueue(e) { //最坏情况O(n)，且可能持续发生

```
    Q.enqueue(e); P.enqueue(e);
    for ( x = P.rear(); x && (e => x->key); x = x->pred ) //见贤
        x->key = e; //思齐
}
```

Queap = Queue + Heap = enqueue + dequeue + getMax = Q + P'

❖ P去重，**升级**至P'：Q中的**关键**元素，才会记录在P'中...



❖ Q.enqueue()和Q.dequeue()，分别对应于计数器的**合**|生、减|灭 ~ 分摊...