

12-C

优先级队列

堆排序

邓俊辉

deng@tsinghua.edu.cn

选取

❖ 在 `selectionSort()` 中

将 **U** 替换为 **H**...

❖ J. Williams, 1964

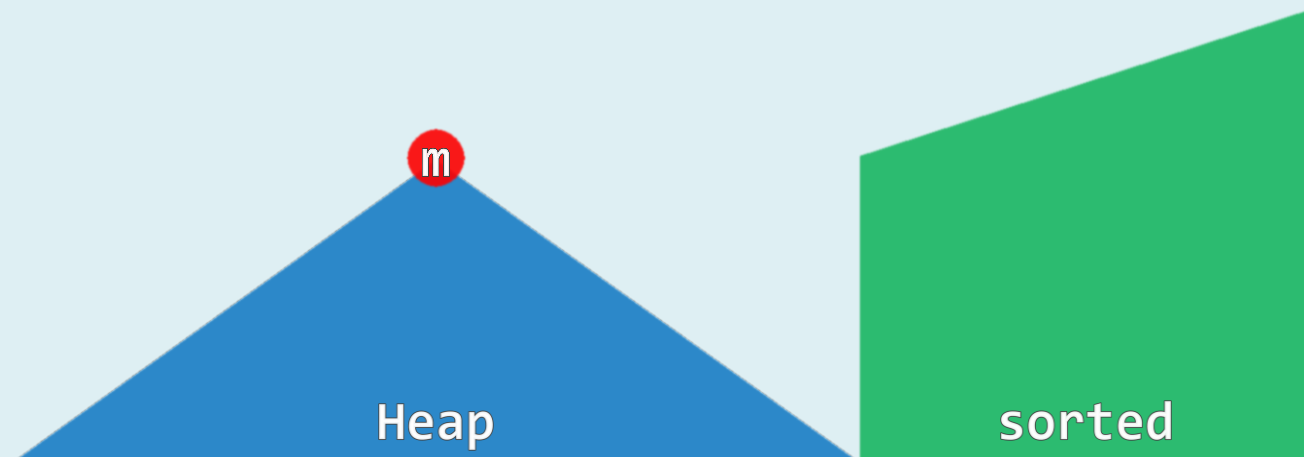
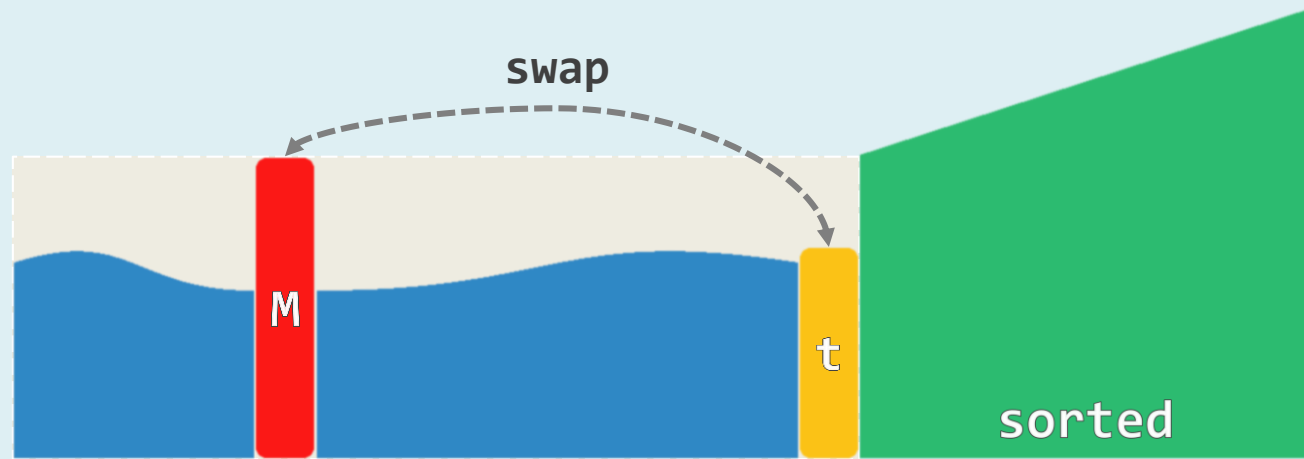
初始化 : `heapify()` , $O(n)$

迭代 : `delMax()` , $O(\log n)$

不变性 : $H \leq S$

❖ $O(n) + n \times O(\log n)$

= $O(n \log n)$



就地

❖ 在物理上

完全二叉堆即是向量

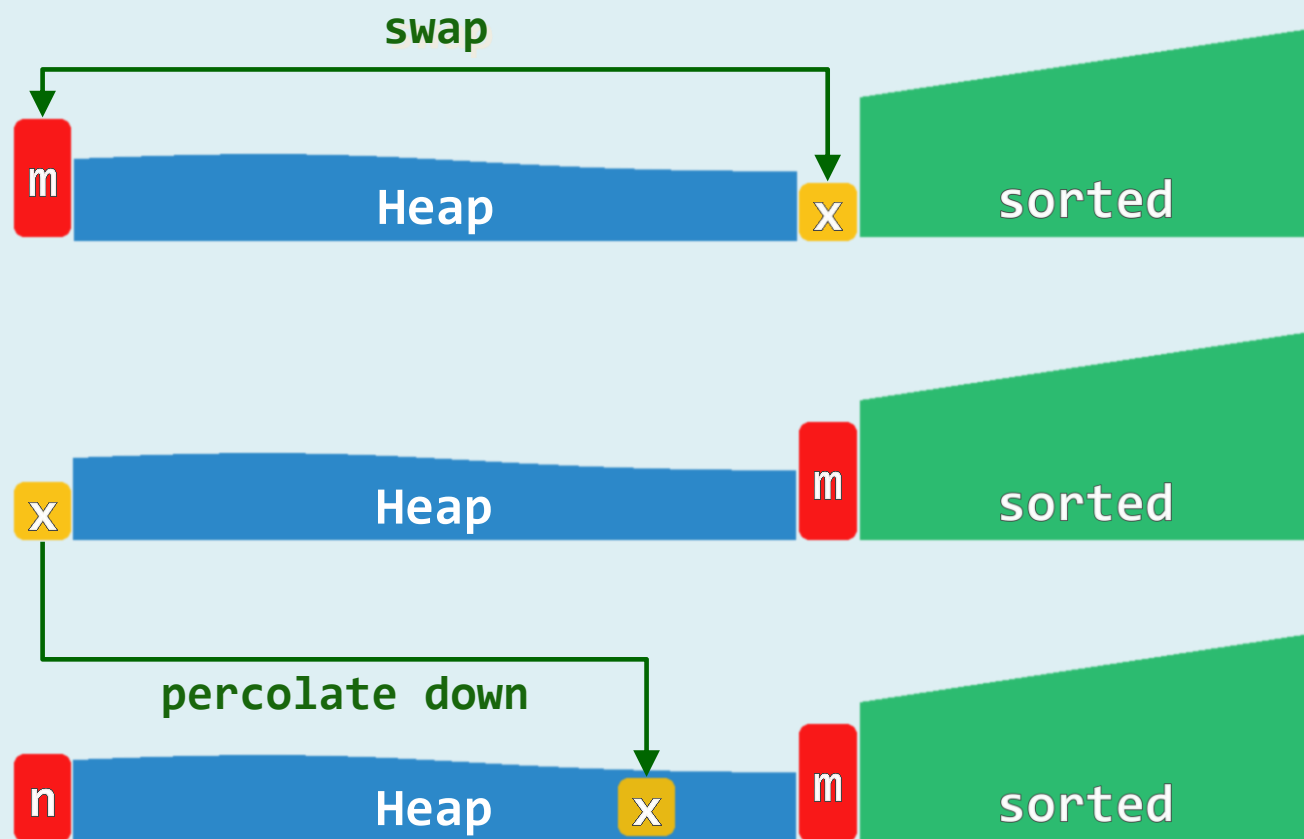
❖ 既然此前有：

$$- m = H[0]$$

$$- x = H[n-1]$$

不妨随即就：

$$- \text{swap}(m, x) = H.\text{insert}(x) + S.\text{insert}(m)$$



实现

❖ template <typename T> //对向量区间[lo, hi)做就地堆排序

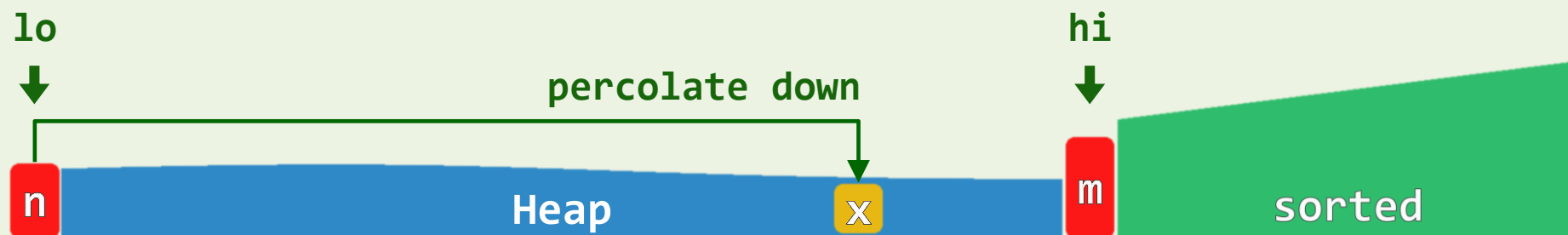
```
void Vector<T>::heapSort( Rank lo, Rank hi ) {
```

```
    T* A = _elem + lo; Rank n = hi - lo; Heapify( A , n ); //待排序区间建堆, O(n)
```

```
    while ( 0 < --n ) //反复地摘除最大元并归入已排序的后缀, 直至堆空
```

```
        { swap( A[0], A[n] ); percolateDown( A, n, 0 ); } //堆顶与末元素对换后下滤
```

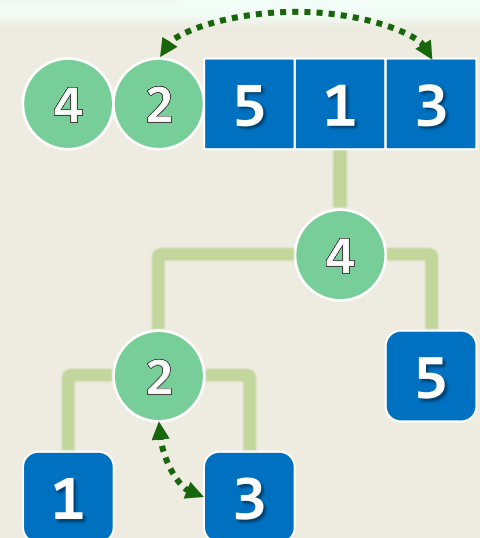
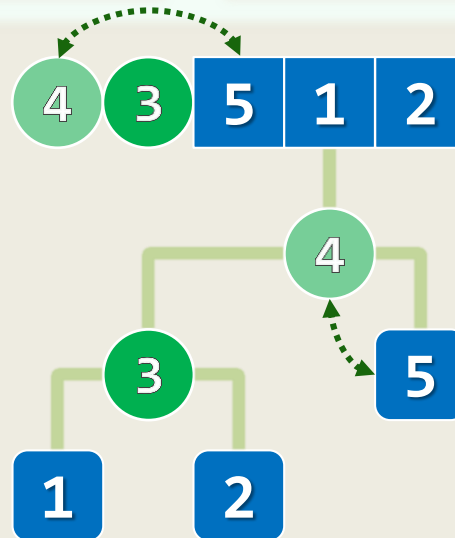
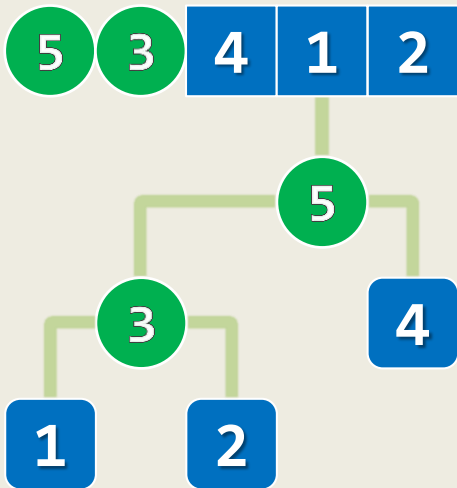
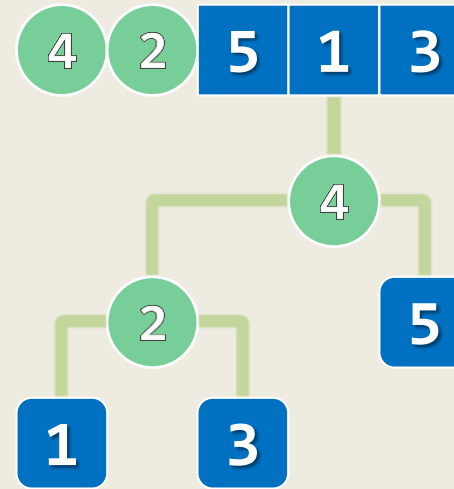
```
}
```



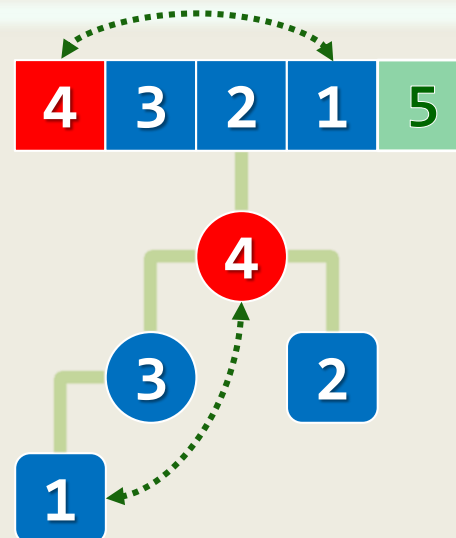
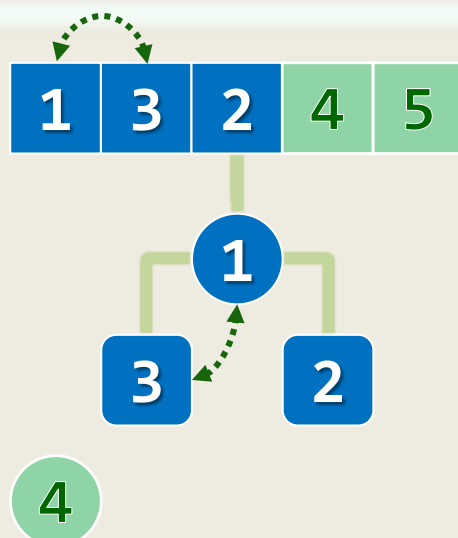
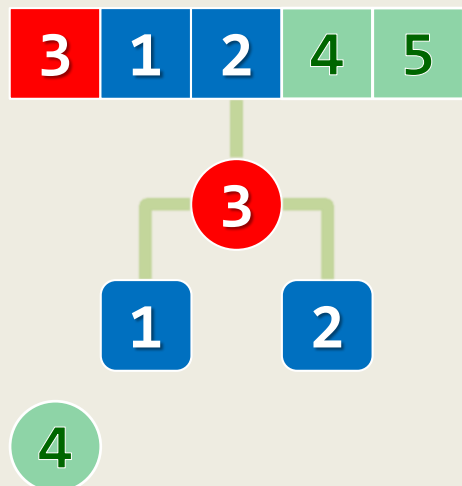
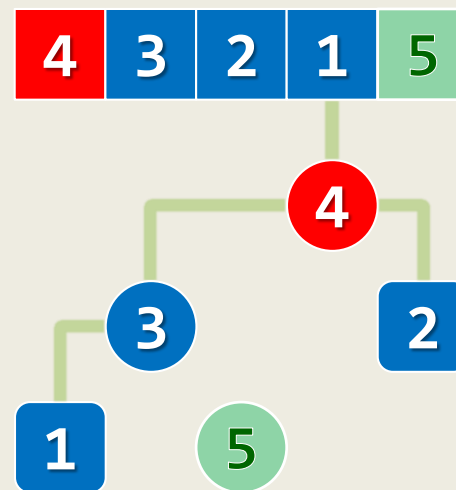
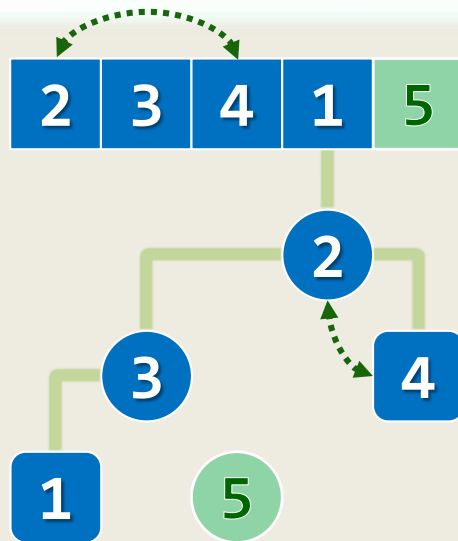
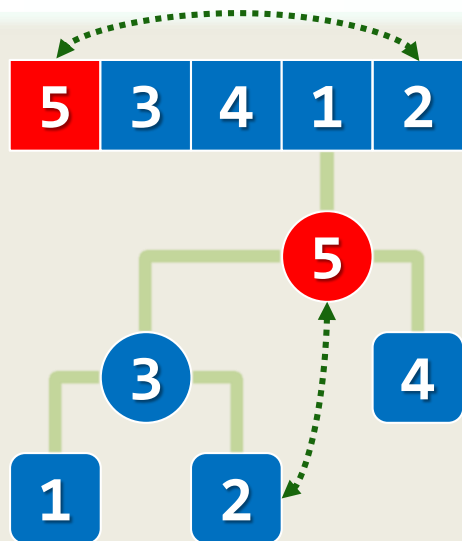
实例：建堆



4 2 5 1 3



实例：选取 + 调整 (1/2)



实例：选取 + 调整 (2/2)

