```
/* ComplHeap.h */
2
    #ifndef COMPL HEAP H
3
    #define COMPL HEAP H
4
5
    typedef int ComplHeapCmp(const void *, const void *);
6
    typedef void ComplHeapFree(void *);
7
    //顺序表表示完全二叉堆
8
9
    typedef struct
10
       void *elems;//顺序表的起始地址
11
       int elemSize;//每个元素的大小
12
       long long size;//当前顺序表元素的个数
13
14
       long long initCap;//顺序表初始容量
       long long capacity;//当前顺序表容量
15
16
       ComplHeapCmp *cmpFn;
17
       ComplHeapFree *freeFn;
18
    } PQUEUE;
19
20
    //优先级队列初始化
21
    void PQueueNew(PQUEUE *pq, int elemSize, long long initCap, ComplHeapCmp *cmpFn,
    ComplHeapFree *freeFn);
    //获取优先级队列元素个数
22
23
    long long PQueueSize(PQUEUE *pq);
24
    //优先级队列判空,返回值:0--非空,!0--空
25
    int PQueueEmpty(PQUEUE *pq);
26
    //清空优先级队列所有元素
27
    void PQueueMakeEmpty(PQUEUE *pq);
    //优先级队列销毁
28
29
    void PQueueDispose(PQUEUE *pq);
    //获取当前优先级最大的元素
30
31
    int PQueueGetMax(PQUEUE *pq, void *e);
    //优先级队列插入关键码e,返回值: 0--成功,!0--失败
32
    int PQueueInsert(PQUEUE *pq, const void *e);
33
    //优先级队列删除优先级最大的元素
34
35
    int PQueueDeleteMax(PQUEUE *pq);
    //批量建堆
36
37
    void PQHeapify(PQUEUE *pq);
38
    #endif
```