9

代码：

void eraseMin () {

//找到并保存最小值的指针

skipNode<K,E>\* to\_delete = search ( headerNode->next[0]->element.first ); //修正跳表结构

for ( int i = 0; i < levels; i++ ) {

if ( last[i]->next[i] != to\_delete ) {

break;

}

last[i]->next[i] = to\_delete->next[i];

}

for ( ; levels > 0;levels-- ) {

if ( headerNode->next[levels] != tailNode ) {

break;

}

}

//长度减一，删除元素

delete to\_delete;

dSize--;

}

//O(n^2)

void eraseMax () {

//找到并保存最大值的指针

skipNode<K, E>\* to\_delete = search ( headerNode->next[levels]->element->first );

//修正跳表结构

for ( int i = 0; i < levels; i++ ) {

if ( last[i]->next[i] != to\_delete ) {

break;

}

last[i]->next[i] = to\_delete->next[i];

}

for ( ; levels > 0; levels-- ) {

if ( headerNode->next[levels] != tailNode ) {

break;

}

}

delete to\_delete;

dSize--;

}

复杂度：两个函数中，查找较慢，复杂度为O(logn)，整理链表复杂度忽略。