/\*

-----------------------------------选择排序----------------------------------

概述：

对数据进行N-1次遍历，每次遍历找出当前最小值（或最大值），并置于起始（终止）位置（这是一个交换操作）

形如：

\*为下一次遍历的起始位置

|为交换操作

8、3、2、4、1、5、7、6、9 --原始数据

\* |

1、3、2、4、8、5、7、6、9 --第一次遍历（并交换）

\* |

1、2、3、4、8、5、7、6、9 --第二次遍历（并交换）

\*

1、2、3、4、8、5、7、6、9 --第三次遍历（并交换）

\*

1、3、2、4、8、5、7、6、9 --第四次遍历（并交换）

\* |

1、3、2、4、5、8、7、6、9 --第五次遍历（并交换）

\* |

1、3、2、4、5、6、7、8、9 --第六次遍历（并交换）

\*

1、3、2、4、5、6、7、8、9 --第七次遍历（并交换）

\*

1、3、2、4、5、6、7、8、9 --第八次遍历（并交换）

复杂度分析：

交换操作：

最佳情况：0次 （譬如：1、2、3、4）

最坏情况：n-1次 （譬如：3、1、4、2）

比较操作：

n-1 + n-2 + ... + 1

= (1 + n-1) \* (n - 1) / 2

= n(n-1)/2

= O（n^2）

\*/

void selectionSort(int\* a, int lo, int hi)

{

for (int result = 0; result < hi; result++) //result：已排好序的结果数组下标

{

int current = result; //当前最小默认为每次遍历的第一个元素

for (int i = result + 1; i <= hi; i++) //i：每次遍历数组的“游标”

if (a[i] < a[current])

current = i;

exch(a, result, current); //交换a[result]和a[current]

}

}