选择排序

基本思想：在待排序的数据中选出最大（小）的元素放在最终的位置

实现思路：

> 第一轮认定第一个值就是最小值，并记录下标为min，然后从剩余的n-1个元素如果有比min下标还小的值，更新min
>
> 如果min最终不是第一个值，第一个值与min指向的值互换，否则不互换
>
> 通过第一轮结束，第一个值为最小值
>
> 第二轮用同样的方式找第二个最小值，以此类推，直到排序结束

选择排序

```
class Select
{
 public:
  void sort(vector<int> &nums)
  {
     int n = nums.size();
    for (int i = 0; i < n - 1; i++)
    {
      int minPos = i;//让当前第i个元素就是最小值
      for (int j = i + 1; j < n; j++)
      {
        //如果后面的值更小，更新最小值下标
        if (nums[j] < nums[minPos])
        {
          minPos = j;
        }
      }
      //互换
      int temp = nums[i];
      nums[i] = nums[minPos];
      nums[minPos] = temp;
    }
  }
};

int main()
{
  vector<int> nums = {9, 7, 5, 3, 1, 0, 8, 4, 6, 2};
  Select select;
```

```
30   select.sort(nums);
31   for (int num : nums)
32   {
33     cout << num << " ";
34   }
35   return 0;
36 }
```

总结：

时间复杂度（平均）：O(n^2)

> 最好：O(n^2)
>
> 最坏：O(n^2)

空间复杂度：O(1)

稳定性：不稳定

适用：数据量少