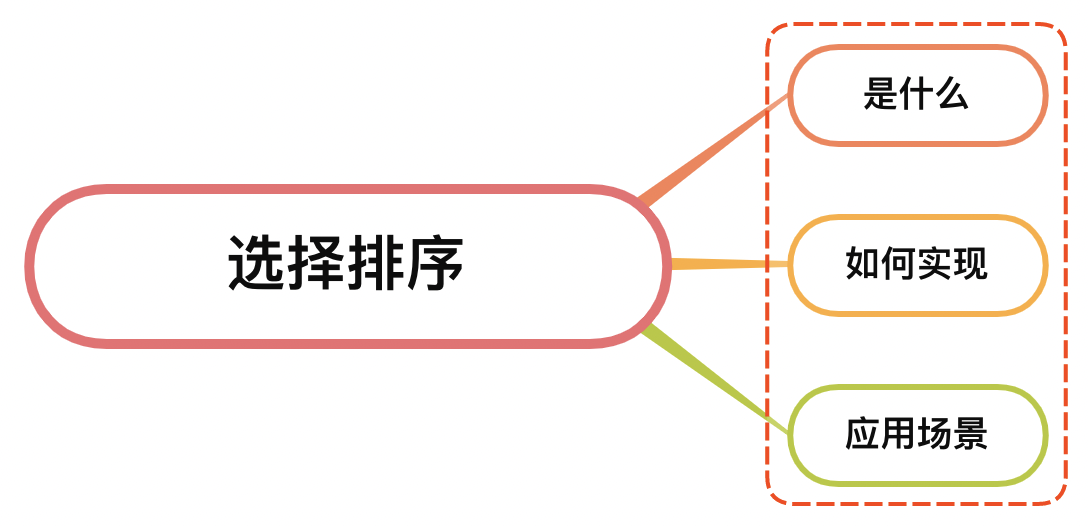
# 面试官：说说你对选择排序的理解？如何实现？应用场景？



## 一、是什么

选择排序（Selection sort）是一种简单直观的排序算法，无论什么数据进去都是 O(n²)的时间复杂度，所以用到它的时候，数据规模越小越好

其基本思想是：首先在未排序的数列中找到最小(or最大)元素，然后将其存放到数列的起始位置

然后再从剩余未排序的元素中继续寻找最小(or最大)元素，然后放到已排序序列的末尾

以此类推，直到所有元素均排序完毕

举个例子，一个数组为 56、12、80、91、29，其排序过程如下：

* 第一次遍历时，从下标为 1 的位置即 56 开始，找出关键字值最小的记录 12，同下标为 0 的关键字 56 交换位置。此时数组为 12、56、80、91、20



* 第二次遍历时，从下标为 2 的位置即 56 开始，找出最小值 20，同下标为 2 的关键字 56 互换位置，此时数组为12、20、80、91、56



* 第三次遍历时，从下标为 3 的位置即 80 开始，找出最小值 56，同下标为 3 的关键字 80 互换位置，此时数组为 12、20、56、91、80



* 第四次遍历时，从下标为 4 的位置即 91 开始，找出最小是 80，同下标为 4 的关键字 91 互换位置，此时排序完成，变成有序数组

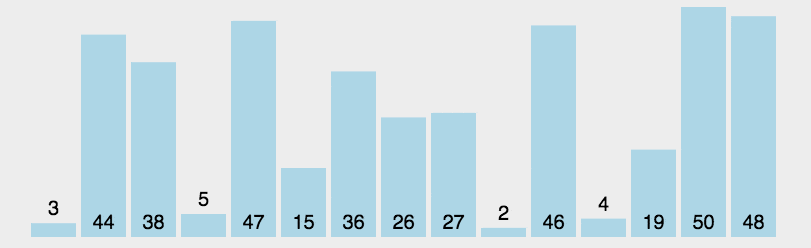


## 二、如何实现

从上面可以看到，对于具有 n 个记录的无序表遍历 n-1 次，第i 次从无序表中第 i 个记录开始，找出后序关键字中最小的记录，然后放置在第 i 的位置上

直至到从第n个和第n-1个元素中选出最小的放在第n-1个位置

如下动画所示：



用代码表示则如下：

function selectionSort(arr) {  
 var len = arr.length;  
 var minIndex, temp;  
 for (var i = 0; i < len - 1; i++) {  
 minIndex = i;  
 for (var j = i + 1; j < len; j++) {  
 if (arr[j] < arr[minIndex]) { // 寻找最小的数  
 minIndex = j; // 将最小数的索引保存  
 }  
 }  
 temp = arr[i];  
 arr[i] = arr[minIndex];  
 arr[minIndex] = temp;  
 }  
 return arr;  
}

第一次内循环比较N - 1次，然后是N-2次，N-3次，……，最后一次内循环比较1次 共比较的次数是 (N - 1) + (N - 2) + ... + 1，求等差数列和，得 (N - 1 + 1)\* N / 2 = N^2 / 2，舍去最高项系数，其时间复杂度为 O(N^2)

从上述也可以看到，选择排序是一种稳定的排序

## 三、应用场景

和冒泡排序一致，相比其它排序算法，这也是一个相对较高的时间复杂度，一般情况不推荐使用

但是我们还是要掌握冒泡排序的思想及实现，这对于我们的算法思维是有很大帮助的

## 参考文献

* https://baike.baidu.com/item/%E9%80%89%E6%8B%A9%E6%8E%92%E5%BA%8F/9762418
* https://zhuanlan.zhihu.com/p/29889599
* http://data.biancheng.net/view/72.html