Sort Algorithm

-Comparison Sort

通过对数组中得元素进行比较来实现排序。利用决策树模型，我们证明任何对规模为n的输入，比较排序算法的最坏情况运行时间的下界是Ω(nlgn)。

heap sort和merge sort都是渐进最后的比较排序算法。

Insertion Sort

-Insertion Sort

Incremental增量策略

思想：当得到一个新元素的时候，保持前面k规模(k<n)的子序列的有序性，将其插入到子序列中。

e.g., 拿牌排序

Runtime: O(n2)

-Shell缩小增量排序

It can be seen as either a generalization of sorting by exchange (Bubble Sort) or sorting by insertion (Insertion Sort).

思想：分组插入排序。组的大小是n/gap，而不是gap。其他的模拟插入排序即可。

Runtime: O(n2). 虽然有三层循环，但实际上第一层和第三层操作的个数结合起来只是n而已。

Selection Sort

类似DP, 动态规划。每次都把问题归纳为相同性质、但规模更小的子问题（k-1个未排好序的子序列）。

-Selection Sort

真的好没技术含量啊。

Runtime: O(n2).

-Heap Sort

思想：建（最大）堆；交换堆顶与堆的最后一个元素；保持堆的性质。

Runtime: O(nlgn)

No extra space needed.

-DP

Exchange Sort

-Bubble

思想：把最大的数往上冒。

Runtime: O(n2)

-Quict Sort

思想：每次选出一个pivot，将序列分为两个子序列，再对每个子序列进行排序。

Runtime: average O(nlgn), worse O(n2)

DC?

-Merge Sort

思想：将一个序列分成两个（或者更多个）等长的子序列，分别排序，然后再将他们合并成排列好的序列。

DC, 分治法。

Runtime: O(nlgn)

Space: extra O(n)

-DC

T(n) = a\*T(n / b) + O(nd)

Master Theorem.

-Linear Sort

线性时间查找。可以这样实现的原因是用空间换了时间，但是这样做要考虑的是，什么情况下能够这样做，i.e., 需要满足什么条件才能这样做。

类似Hash表。

这是pseudo-polynomial吗？

-Counting Sort

思想：统计字频（这样的前提条件是，开一个0~k大小的数组，把元素当成数组下标）->累加字频，变成在此之前有几个元素->从后往前将元素放到新的数组中。

受限的地方是，1.整数；2.假如输入的最大的数是9999，那会有很大的空间浪费。

Runtime: O(n+k) / O(n)

-Radix Sort

思想：计数排序的进化版，不过还是只能针对整数进行操作。对每一位（从低到高）进行计数排序。

Runtime: O(k\*n), k是位数。

-Bucket Sort

思想：a general version of Radix Sort.只不过区间是认为定的而不再是10。

Runtime: O(k\*n), k是桶的个数。

没实现了，太懒了。