people = sorted(people, key = lambda x: (-x[0], x[1]))

排序：表示对people列表进行排序，按照第一个元素降序，第二个元素升序的规则，第一个元素是主要关键字，第二个元素是次要关键字

task\_sort = sorted(task\_map.items(), key=lambda x: x[1], reverse=True) 对字典按value从大到小进行排序

注意一下背包问题，0-1背包是每个元素只能选择一次，完全背包问题是每个元素可以选无数次，这种元素放进背包——即几个元素和、相连接能否组成xx、组成xx有几种办法都可以转换成背包问题，最终也就是用动态规划来解。

涉及连续子数组和：前缀

如果不要求连续：背包

如果a，b是数值变量， 则&， |表示位运算， and，or则依据是否非0来决定输出

如何a, b是逻辑变量， 则两类的用法基本一致

KMP <https://www.zhihu.com/question/21923021>

本质：模式串P像标尺一样在主串上滑动，暴力搜索的话每一次就滑动一步，KMP每次滑动多步，避免 不可能成功的尝试

[]内装索引 ()内装函数的参数

当碰到需要避免自身匹配自身的情况，可以先判断能否匹配，再加入元素(股票问题，两数之和)

注意快排quicksort(nums,left,right)的第一个判断条件 if left<right

回溯：注意能否重复选取元素以及最后结果是否需要去重

字典按值的大小进行排序,返回的是一个列表 **sorted(dict.items(), key=lambda e:e[1], reverse=True)**