1. 功能上

第一个函数take用于从列表前面取出前 i 个元素组成新列表，例如  
take([1,2,3,4，5], 3) => [1,2，3]

第二个函数rtake用于从列表前面取出前 i 个元素**逆序**添加到taken之前，例如  
rtake([1,2,3,4], 2, []) => [2,1]

性能上

第一个函数take为传统递归，调用栈层数为i，而第二个函数rtake为尾递归，空间复杂度低，性能更好

2.

最终答案为**nextperm[2,3,1,4] => [3,1,2,4]**

计算过程如下：

nextperm[2,3,1,4]

=> next([2], [3,1,4])

(\*2<3? true\*)

=> next([3,2], [1,4])

(\* 3<1? false \*)

=> swap([3,2]) (\* xlist=[3,2], y=1, ys=[4] \*)

=> 匹配第二个模式：x::xk::xs，其中 x=3, xk=2, xs=[]

=> (\* 检查 xk=2 > y=1? 是 \*)

=> x::swap([2]) (\* 递归调用 swap([2]) \*)

swap([2]) (\* xlist=[2], y=1, ys=[4] \*)

=> [x](\* 匹配第一个模式,其中 x=2 \*)

=> y::x::ys = 1::2::[4] = [1,2,4]

=> 3::swap([2]) = 3::[1,2,4] = [3,1,2,4]

3.

见附件，该函数不保证找到的子集唯一，只能输出一种符合要求的子集或NONE