Question: https://leetcode.com/problems/remove-k-digits/

So in order to get lowest number combination the number's digit should be all in ascending order and the left most digits contain higher weightage so we will use stack to implement this.

We will push items to stack if their previous elements or top of stacks are smaller than them, we will do this k times.

And if we cannot do it k times, lets say that the digits are already in ascending order then we just remove the last k digits.

Code:

class Solution {

public String removeKdigits(String num, int k) {

if(num.length()==k){

return "0";

}

Stack<Character> stack = new Stack<>();

for(int i=0; i<num.length(); i++){

Character c = num.charAt(i);

while(!stack.empty()&&c<stack.peek()&&k>0){

stack.pop();

k--;

}

if(stack.empty()&&c!='0'){

stack.push(c);

continue;

}

if(!stack.empty()){

stack.push(c);

}

}

while(k!=0&&!stack.empty()){

stack.pop();

k--;

}

String res="";

while(!stack.empty()){

res=stack.peek()+res;

stack.pop();

}

if(res==""){

res="0";

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>