/\*

Given a list of non negative integers, arrange them such that they form the largest number.

For example, given [3, 30, 34, 5, 9], the largest formed number is 9534330.

Note: The result may be very large, so you need to return a string instead of an integer.

方法一：（没写出来）

用map来存 ，每一个数后面加一个k，是为了8优先于80

这种情况！

注意：map是从小的存到大，逆序遍历需要

map<string,int>::reverse\_iterator it=pp.rbegin();

map<string,int>::reverse\_iterator it\_end=pp.rend();

for(;it!=it\_end;it++)

方法二：冒泡排序

\*/

class Solution {

public:

bool is\_change(string a,string b)//如果a<b则交换

{

string ab=a+b;

string ba=b+a;

if(ab<ba)

return 1;

else

return 0;

}

string largestNumber(vector<int>& nums)

{

//先用map的特性排一次序

map<string,int> pp;

for(int i=0;i<nums.size();i++)

{

string tt= to\_string(nums[i]);

pp[tt]++;

}

string ret="";

map<string,int>::reverse\_iterator it=pp.rbegin();

map<string,int>::reverse\_iterator it\_end=pp.rend();

vector<string> num;

for(;it!=it\_end;it++)

{

for(int i=0;i<it->second;i++)

num.push\_back(it->first);

}

//再用冒泡排序

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

{

for(int j=0;j<num.size()-i-1;j++)

{

if(is\_change(num[j],num[j+1]))

swap(num[j],num[j+1]);

}

}

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

ret+=num[i];

//如果全是0则返回"0"

if(ret[0]=='0')

return "0";

return ret;

}

};