Pancake Sort： Code implementation

Pseudocode:

Len=len[A]

for(0->len)  
 index=(max(A[0->(len-i)]).index  
 res.add(index)  
 filp0-index  
 filp0->(len-i)  
 if sorted:

return res  
 return res

C++ implementation:

#include <iostream>

#include <vector>

#include <cmath>

//A is the array to be sorted and n is the length

vector<int> pancakeSort(int[] A，int n ) {  
 vector<int> res;  
 int len = n;  
 int maxIndex = 0;  
 for(int i = 0;i<len;i++){  
 maxIndex = findMax(A,len-i);  
 res.push\_back(maxIndex);  
 overTurn(A,maxIndex);  
 res.push\_back(maxIndex)  
 overTurn(A,len-i);  
 if(isOK(A,len-i-1)) return res;  
 }  
 return res;  
 }  
   
 int findMax(int[] A,int len){  
 *//Find max*

int max =INT\_MIN  
 int index = 0;  
 for(int i = 0;i<len;i++){  
 if(A[i]>max) {  
 index = i;  
 max = A[i];  
 }  
 }  
 return index+1;  
 }  
void overTurn(int[] A,int len){  
 int temp = 0;  
 for(int i = 0;i<len/2;i++){  
 temp = A[len-i-1];  
 A[len-i-1] = A[i];  
 A[i] = temp;  
 }  
 }  
bool isOK(int[] A,int len){  
 for(int i = 0;i<len-1;i++){  
 if(A[i]>A[i+1]) return false;  
 }  
 return true;  
 }  
}