Climbing Stairs

最开始还没有反应这是一个动态规划问题，导致错误思路，以为是组合问题，最开始全1，后来任意找两个1组成2，提交了如下代码：

class Solution {

public:

int climbStairs(int n) {

int m,sum,i;

sum=1;

if(n==1) return 1;

if(n==2) return 2;

m=floor(n/2);

for(i=0;i<m;i++)

{

sum=sum\*2;

}

if(n%2==0) return sum+1;

else return sum+max(1,m);

}

};

后来意识到是动态规划问题时，得到式子：C(n)=C(n-1)+C(n-2)；采用递归，又忽略了时间，导致超时，提交了如下代码：

class Solution {

public:

int climbStairs(int n) {

if(n==0) return 0;

if(n==1) return 1;

if(n>1) return climbStairs(n-1)+climbStairs(n-2);

}

};

最后用数组来保存每一个值，解决问题，代码如下：

class Solution {

public:

int climbStairs(int n) {

int a[n];

int i;

if(n==1) return 1;

if(n==2) return 2;

if(n>2)

{

a[0]=1;

a[1]=2;

for(i=2;i<n;i++)

{

a[i]=a[i-1]+a[i-2];

}

return a[i-1];

}

}

};