今天在编写dfs的程序的时候，我突然被这个算法的终止条件即：if(i==n)return sum==k;搞糊涂了，我认为这个递归函数只会执行一次就会返回而退出执行。到底是不是这样的呢？这个函数会搜索所有的可能性吗？

经过在纸上的推演，我可以肯定的是，这个函数不会仅仅只执行一次就退出，因为它是二枝的，它有两个分支。如果第一个分支没有找到解，即返回值为false，那么它就不会执行return true这条语句，这种情况下，函数是不会返回而退出执行的，就会接着执行第二个分支。

还有需要抓住一点的是，递归函数的执行是在栈中的，也就是说，每一次调用下一个递归函数就会将这个递归函数压入栈中，所以，更加深层的递归函数肯定会先返回，高层的递归函数则根据底层的递归函数的返回值来决定最终的返回值是什么。这一点是需要格外注意的。

int n,k;

int a[MAX\_N]

bool dfs(int i,int sum){

if(i==n)return sum==k;

if(dfs(i+1,sum)return true;

if(dfs(i+1,sum+a[i]))return true;

return false;

}

void solve(){

if(dfs(0,0)printf("Yes\n");

else printf("No\n");

}

因为现在我的电脑上没有c++的编译环境，所以我首先在纸上来分析一下这个函数的执行，对于上面的这种二叉树型的递归函数的纸上分析，最好可以采用树的形式，画出递归调用树，然后采用二叉树的先根遍历来进行遍历，就可以得到函数的调用过程。