/\*\*

\* BinarySearch.c 折半查找法(递归实现)

\*

\* @author DBL 970073804@QQ.COM

\* @copyright 2018-2019 DBL

\*

\*

\*/

#include <stdio.h>

int binary\_search(int str[],int low,int hight,int key)

{

int mid;

if(low>hight)return -1;

mid = (low+hight)/2;

if(str[mid]==key)return mid;

if(str[mid]>key)return binary\_search(str,low,mid-1,key);

if(str[mid]<key)return binary\_search(str,mid+1,hight,key);

}

int main()

{

int k,str[11] = {1,1,2,3,5,8,13,21,34,55,89};

int addr;

printf("Please enter a key you want to search:\n");

scanf("%d",&k);

addr = binary\_search(str,0,10,k);

printf("The addr is :%d\n",addr);

return 0;

}

时间复杂度:O(log2n)

　　假设对n个元素的折半查找需要消耗的时间为t(n)。容易知道：

　　如果n = 1，则t(n) = c1

　　如果n > 1，则t(n) = t(n/2) + c2

　　其中n/2需要取整，c1、c2都是常数

　　对于正整数n，可以有：

　　t(n) = t(n/2) + c2

　　= t(n/4) + 2\*c2

　　= t(n/8) + 4\*c2

　　= ...

　　= t(n/(2的k次方)) + k\*c2

　　一直推演下去，直到n/(2的k次方)等于1，也就是k = log2(n)，此时等式变为：

　　t(n) = t(1) + k\*c2

　　= c1 + log2(n)\*c2

于是时间复杂度为log2(n)。注意log2(n)和log(n)其实是同样的复杂度，因为它们之间仅仅差了一个常量系数而已。