# 1.宝石与石头

给定字符串J 代表石头中宝石的类型，和字符串 S代表你拥有的石头。 S 中每个字符代表了一种你拥有的石头的类型，你想知道你拥有的石头中有多少是宝石。

J 中的字母不重复，J 和 S中的所有字符都是字母。字母区分大小写，因此"a"和"A"是不同类型的石头。

****示例 1:****

****输入:**** J = "aA", S = "aAAbbbb"****输出:**** 3

****示例 2:****

****输入:**** J = "z", S = "ZZ"****输出:**** 0

****注意:****

* S 和 J 最多含有50个字母。
* J 中的字符不重复。

第一天的第一题的通过率最高的题目，emmmmm、

1.python

class Solution(object):

def numJewelsInStones(self, J, S):

"""

:type J: str

:type S: str

:rtype: int

"""

a=0

for i in J:

a=a+S.count(i)

return a

count函数还是不错滴



2.python

class Solution(object):

def numJewelsInStones(self, J, S):

"""

:type J: str

:type S: str

:rtype: int

"""

s1=0

i=0

while i<len(S):

x=0

while x<len(J):

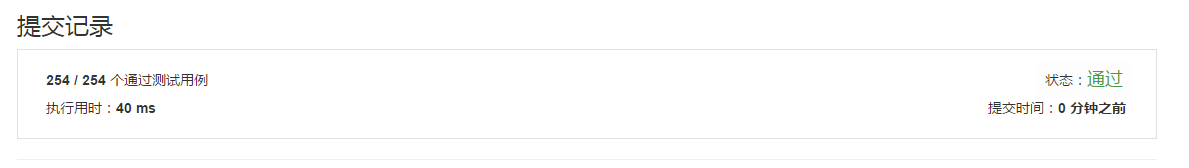
if J[x]==S[i]:

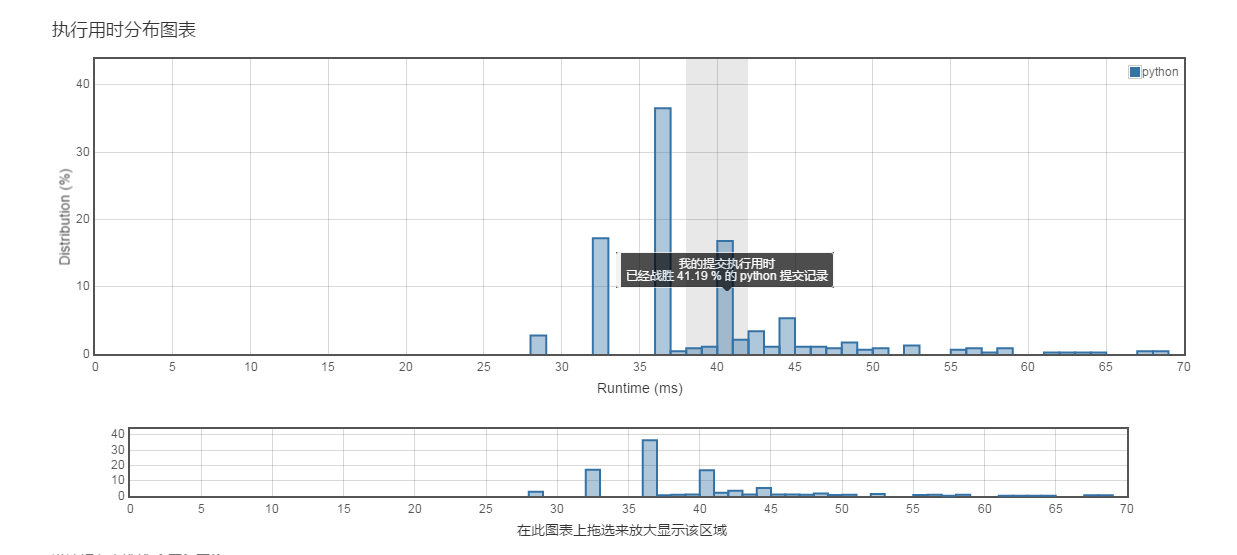
s1=s1+1

x=x+1

i=i+1

return s1





3.c#

public class Solution {

public int NumJewelsInStones(string J, string S) {

int i = 0;

int x = 0;

int a = 0;

for(i=0;i<S.Length;i++){

for(x=0;x<J.Length;x++){

if(J[x]==S[i]){

a= a+1 ;

}

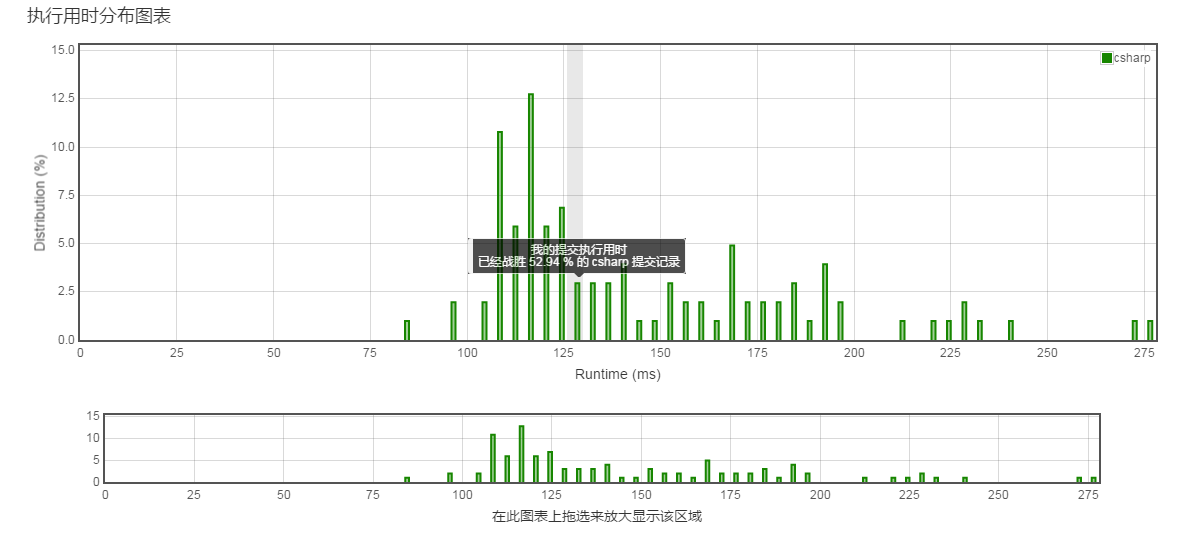
}

}

return a;

}

}



4.c#哈希表= =

public class Solution {

public int NumJewelsInStones(string J, string S) {

int res = 0;

//1、将宝石添加到hashtable 里

char[] ch = J.ToCharArray();

Hashtable hs = new Hashtable();

for (int i = 0; i < ch.Length; i++)

{

hs.Add(ch[i],i);

}

//2、判断hashtable 是否包含

char[] chs = S.ToCharArray();

for (int i = 0; i < chs.Length; i++)

{

if (hs.ContainsKey(chs[i]))

{

res++;

}

}

return res;

}

}