<https://github.com/Milo-dd/sizeYS.git>

### 题目要求：

* 能自动生成小学四则运算题目（注意是给小学生用的，要是结果出现负数的话他们会迷茫的！）
* 除了整数外，还要支持真分数的四则运算

### 目标：

1. 首先该项目是给小学生用的，初步采取交互式对话方式来实现小学生做四则运算的题目。
2. 四则运算的出题范围必须是小学生理解的范围内，不可以超纲（如负数的计算，无理数计算，非真分数计算，高位小数计算。）；
3. 面对不同年纪的小学生可以出不同的题目，(如一年级的学生不能实现分数除法，但是可以做100以内比较简单的带余除法)；
4. 在交互式对话做题的上面可以实现统计小学生做每道题目的用时，并且利用聚类算法来归类分析给老师可以了解每个学生的强项和弱项、和做题的集中度等方法；
5. 根据4的出的数据在人工指导的情况下将小学生计算能力分级，方便老师因材施教和有教无类；
6. 具有出题的功能，可以根据需求出N道题目来给老师或者学生使用。

### 思路：

1. 分数newz()和整数newf()的四则运算是分开的两个函数。利用random随机选择函数生成；
2. 用列表存放四个运算符号，根据选定的运算符号来决定是生成哪种运算；
3. 不能出现负数所以每对随机生成数A和B都必须是大的在前面；
4. 除法是比较特殊的地方根据小学生的年纪决定，1、2年纪的学生只会带余数的除法。再往后的话可以设计真分数的运算，所以除法生成数的A、B顺序根据学生的个人能力来衡量；
5. 分数方面利用Fraction库来实现分数的加减乘除；
6. 交互式做题中利用时间差来统计该题目小学生停留的时间；
7. 利用matlab库来进行数据挖掘，聚类；
8. 利用多次调用的方法快速生成题目。

### 开发实现过程：

关键函数：

def newf(): #分数生成

def newz():#整数生成

def xianzuo():#交互式题目生成

def shengcheng():#迅速生成题目

实现过程：

用户输入1，2，00来选择：交互式做题、快速生成题库、退出

* 在交互式做题 输入答案会先评价答案是否正确，再生成下一题，如果输入00则直接退系统。
* 快速生成题库会迅速生成用户想要的题目数量，答案和题目是分开的。
* 退出。直接退出

### 代码说明：

以下为关键函数的描述，其余不作赘述：

def newz()

|  |
| --- |
| def newz():  f=['+','-','×','÷']  num=random.randint(0,3)  add1=random.randint(1,20)  add2=random.randint(1,20)  res=[]  if add1<add2:  temp=add1  add1=add2  add2=temp  if num==0:  val=str(add1+add2)  elif num==1:  val=str(add1-add2)  elif num==2:  val=str(add1\*add2)  elif num==3:  if add1>add2 and add1%add2!=0:  temp = add1  add1 = add2  add2 = temp  val=str(Fraction(add1,add2))  res.append(val)  res.append(str(add1) + ' '+f[num]+' ' + str(add2))  return res |

def newf():

|  |
| --- |
| def newf():  f=['+','-','×','÷']  num=random.randint(0,3)  t1 = random.randint(1, 20)  t2 = random.randint(t1, 20)  a = Fraction(t1, t2)  t1 = random.randint(1, 20)  t2 = random.randint(t1, 20)  b = Fraction(t1, t2)  res=[]  if num==0:  val=str(a+b)  elif num==1:  if a<b:  temp = a  a = b  b = temp  val=str(a-b)  elif num==2:  val=str(a\*b)  elif num==3:  val=str(Fraction(a,b))  res.append(val)  res.append(str(a) + ' ' + f[num] + ' ' + str(b))  return res |

def xianzuo():

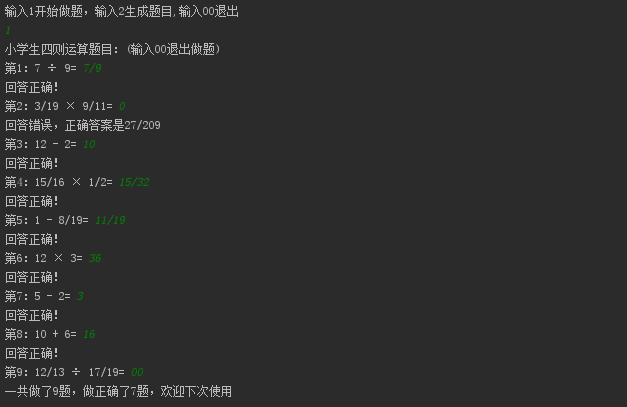
|  |
| --- |
| def xianzuo():  ok = 0  i = 1  r = 0  print("小学生四则运算题目：(输入00退出做题)")  while ok != '00':  t = random.randint(0, 1)  if t == 0:  R = newz()  else:  R = newf()  print("第%d：%s=" % (i, R[1]), end=" ")  ok = input()  if ok == '00':  break  elif ok == R[0]:  print("回答正确！")  r += 1  else:  print("回答错误，正确答案是%s" % R[0])  i += 1  print("一共做了%d题，做正确了%d题，欢迎下次使用" % (i, r)) |

def shengcheng():

|  |
| --- |
| def shengcheng():  n=input("生成题目多少道：")  list1=[]#保存题目  list2=[]#保存答案  for i in range(int(n)):  t = random.randint(0, 1)  if t == 0:  R=newz()  list1.append(R[1])  list2.append(R[0])  else:  R = newf()  list1.append(R[1])  list2.append(R[0])  #print(R[1]+"="+R[0])  #print(type(list1[0]))  for o in range(len(list1)):  #print(type(i))  if o%5==0:  print("\n")  print("%d、%s= " % (o + 1, list1[int(o)]), end=" ")  #print("%d."+list1[o]+"=",end=" ")  print("\n答案如下：")  for j in range(len(list2)):  if j%5==0:  print("\n")  print("%d、%s " %(j+1,list2[int(j)]),end=" ")   print("\n生成完毕，一共%s道题目"%n) |

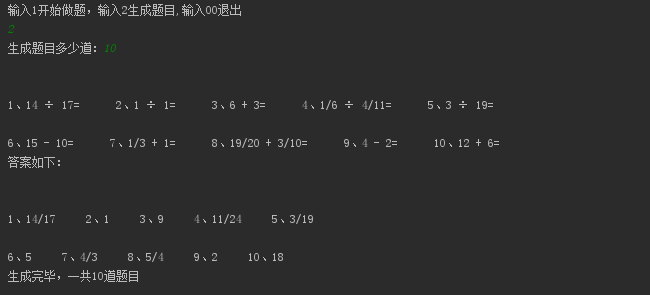
### 测试运行：

交互式做题测试如下



由于时间关系我只做了少量题目的测试。

快速生成题目的测试：



在这个简单的测试中发现了一个bug，当两个真分数相加时出现了假分数5/4。不知道这个是否是小学生理解的范围内。

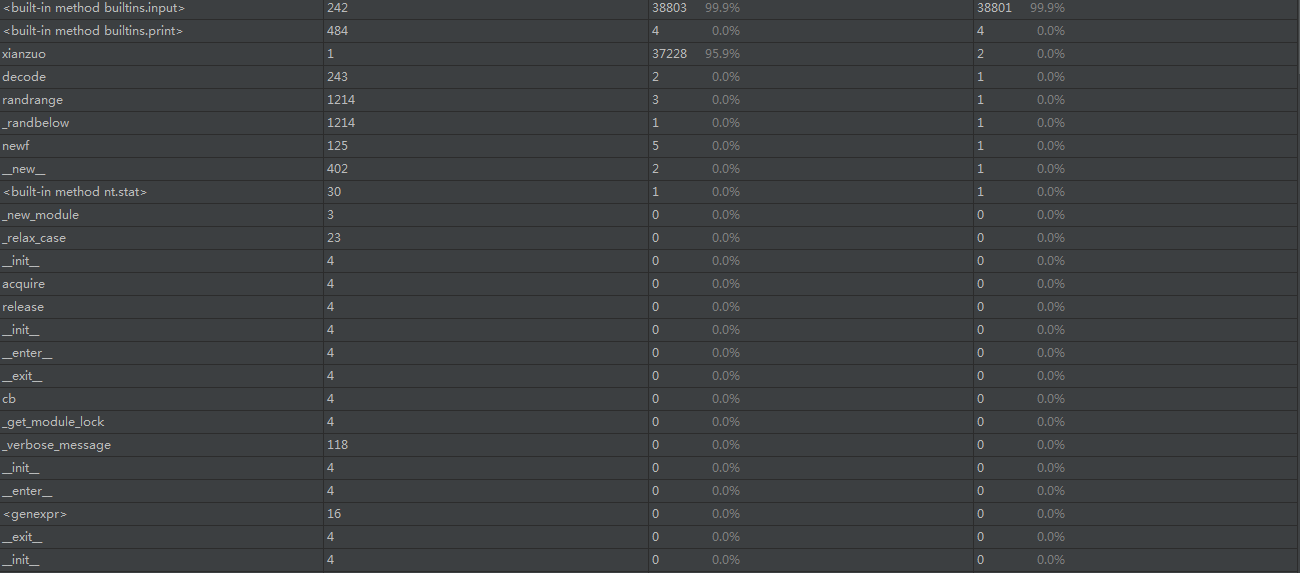
### 效能分析：

PyCharm提供了性能分析工具Run——Profile

当调用快速生成并生成10000条题目时：

### 

当选择交互式做题时：



显然当调用的函数不同时他们占有的时间就不一样。

但我确实不知道该如何优化我的函数，能力有限留下次做

### 项目评价：

由于我开始做这个项目的时间比较晚所以我没有做出一个完整的软件。

以下几点可以留做日后改进的备忘录：

* 首先对象是小学生，所以应该做一个图形用户界面；
* 其次没有实现数据挖掘和分析的功能；
* 做一个实现错题记录的功能；
* 做一个分数评测功能；
* 做一个随堂测试功能；
* 做一个评分功能；

|  |  | 预计时间 | 需要时间 |
| --- | --- | --- | --- |
| Planning | 计划 | 10min | 30min |
| Estimate | 估计这个任务需要多少时间 | 8h40min | 400min |
| Development | 开发 | 40min | 90min |
| Analysis | 需求分析 | 1h | 30min |
| Design Spec | 生成设计文档 | 1.5h | 0 |
| Design Review | 设计复审（和同事审核设计文档） | 30min | 0 |
| Coding Standerd | 代码规范（为目前的开发制定合适的规范） | 20min | 0 |
| Design | 具体设计 | 30min | 0 |
| Coding | 具体编码 | 1h | 33min |
| Code Review | 代码复审 | 30min | 24min |
| Text | 测试（自测，修改代码，提交修改） | 20min | 15min |
| Reporting | 报告 | 40min | 28min |
| Text Report | 测试报告 | 20min | 10min |
| Size Measurement | 计算工作量 | 10min | 5min |
| Postmortem & Process Improvement Plan | 事后总结，并提出过程改进计划 | 1h | 2h |