

算法实现原理：

- 从磁盘上装载Factor和value的文件中读入所有的信息
- 设置 参数 t 和m, 以及预计这一次计算的时常 run_hour
- 算法初始化数据结构, 比如自动生成未被覆盖的 t-way combination集合, 记录所有的factor以及每个factor的每一个value在所有 t-way combination中出现的次数 (因此对应于论文中AETG算法的第一步, 在这种实现中是O (n) 时间复杂度的)
- 执行以下代码:

```
1 while(存在未被覆盖的t-way combination):
2     计算已经花费的时间, 如果大于预设时间run_hour:
3         将中间结果序列化到文件中
4         break
5     生成M个testCase, 选择之中能够覆盖最多 t-way combination的testCase, 加入到结果集中
6     更新未被覆盖的 t-way combination集合。
7     如果上述集合空了:
8         将结果写到./result.csv中
9         计算打印算法花费的总时间
```

心得体会：

- 感觉如果用 Java 写, 在总时间开销表现上会好很多, 而且写起来也不会感觉特别奇怪
- 这个作业很有综合性, 让我对python更熟悉了

代码如何运行：

下面的链接中有配合源代码的详细的注释以及运行指导, 以及最后一次运行的输出记录

https://colab.research.google.com/drive/1se5cHoEpEIFdNvffQS_sTjk7byFxVd-f?usp=sharing