

# Homework 11: Cuckoo Hash

---

本次作业需要你自己实现一个 Cuckoo Hash，并测试其并行与串行的效率。关于 Cuckoo Hash 的内容请参考 Slides。

## Part 1: Cuckoo Hash 实现

---

1. 请实现一个 Cuckoo Hash；

具体的实现代码请参考 Canvas 上的 [第十章Cuckoo hash.pdf](#) 的 10.3-10.5 节；（你也可以自行实现）

2. 需要注意的是，上述参考中给出的实现并没有包括陷入死循环之后的 Rehash 的内容，这部分需要你自已实现，一个解决方法是**路径检测+重新哈希**：

- 检测循环路径的方法也比较简单，可以预先设定一个较大的**阈值**，当循环次数或者递归调用次数超过设定阈值时，就可以认为产生了循环路径。
- 一旦判断发生循环路径之后，进行 Rehash 操作，选择**新的 Hash 函数**，然后**重新构造整个哈希表**。

## Part 2: 加速比竞赛

---

本实验需要测试并比较 Cuckoo 在不同**串行与并行**情况下插入与查找的性能，看看你能否取得更高的加速比。

竞赛中你需要：

1. 以哈希表容量为 **100000** 为例，在插入键值对数量为 **10000**，**50000** 的情况下，记录**串行**插入数据的时间和**串行**查找时间；
2. 采用**并行**插入和**并行**查找的方式重新实验，记录时间；
3. 计算你的实验中的两种操作各自的并行处理的加速比（**并行时间 / 串行时间**），简单分析实验结果是否与 CPU 个数成正比，导致这样结果可能的原因。

Tips:

- 测试的键值对可以只使用键来模拟，即键与值都是相同的数字；
- 测试集自行选取，查找只考虑成功的情况即可；
- 根据 Part 1 中提供的文档，并行的实现只需要对 `Cuckoo::put` 函数有所修改，可以参考 10.5 小节 **并行put** 的内容，具体的并行测试的方法也可以参考文档的 10.4.2 小节的代码；

## Part 3: 提交要求

---

你提交的内容应该包括：

- 你实现的 Cuckoo Hash 的源代码；
- 你进行加速比竞赛中**测试**的源代码；
- 你的**实验结果数据**以及对实验结果的**简单分析**；

## Part 4: 注意事项

---

- 请将相关的代码和实验报告打包上传 Canvas，命名使用“学号+姓名+hw11”，如“521123456789+张三+hw11.zip”。
- **请勿抄袭！** 课后作业采用倒扣分制，如果有遗漏或者得分不足会在最终成绩酌情减分，同时课后作业的内容会体现在期末试卷中，对同学们也是一种练习。
- 本次作业的截止时间是 **2023年5月21日23:59**，迟交将会酌情扣分。
- 有任何作业相关的问题可以询问 江珣璠、熊天磊 助教