

# 项目1：量子模拟器加速

---

## 项目地址

---

<https://github.com/lhclbt/ZJU-summer2025/tree/main/quantum-simulator-speedup>

## 流程说明

---

根据本文档和introduction.ipynb中的介绍和引导，完成项目，编程语言不限。

最后需要提交任务报告，并准备最后阶段的汇报。

围绕任务设计实验，把一切所思所想和实验结果写进报告；即使因为时间关系没有完成任务或效果不尽理想，也建议把所有所做的工作写进报告。

## 文件说明

---

1. introduction.ipynb是可交互的引导文件，包含示例python代码。
2. circuit.py 是实际的项目代码，本项目将完善该框架以完成目标。
3. python环境为3.10
4. requirement.txt 内是项目依赖包, 可以 " pip install -r requirements.txt" 安装或自己手动安装，建议使用 anaconda/uv等环境管理工具避免出现版本冲突或各类疑难杂症。**注意qiskit版本为1.2，不是最新版，切勿装错引发未知冲突**
5. introduction.ipynb最后一节的门依赖关系可视化，需要安装graphviz的二进制文件。请参考<https://graphviz.org/download/> 完成安装

## 任务目标

---

完善circuit.py，加快circuit.matrix()的效率，可以使用任何编程语言（python、matlab、C、R，CUDA等），可以使用多进程，分布式等任意方法

## 需要提交的材料

1. 可执行代码
2. 代码介绍，可以是word、pdf、markdown形式,需包含：
  1. 实现介绍
  2. 已有的加速实现方式 + future work（不需要实现）
  3. 测试结果，包含了测试的硬件、软件、电路的规模（比特数、门数量）、计算时间、baseline（本项目提供的代码）在同一台电脑上执行同一个电路计算的时长
3. 你需要准备向我们解释和介绍你的成果，时长约15分钟。

请不要作弊，我们会根据你所描述的实现方式预测加速情况，如果测试不合理我们会要求进行一对一的验证

## 建议与参考

---

### 部分优化思路参考

1. 提高代码的并行度，利用并行计算相关的知识；
2. 减少重复计算；
3. 利用一下稀疏矩阵的计算方法进行加速；
4. 研究一下**tensor product**和**permute**函数，发现其中的规律（我们可以把它从两个 $(2N, 2N)$ 的矩阵乘法变成一个 $(2N, 2N)$ 和 $(2,2)$ 的矩阵的乘法）；
5. 改成蒙特卡洛采样的方法
6. 找到一个电路优化方法，减少需要优化的电路；
7. 互相之间不依赖的门可以并行的计算
8. ....

### 部分参考推荐与说明

项目代码框架，introduction.ipynb 内的演示都使用了qiskit，这是IBM公司开发的非常常用的量子计算开源库。但由于其对中国IP封锁，官方在线文档<https://docs.quantum.ibm.com/guides>需要全局非大陆位置代理（香港不行）才能访问，最好是北美节点。

注意勾选版本为1.2.x，我们使用的并非最新版。

对于本项目，了解qiskit本身并非必需。