Guideline:

1. 结合paper（见 note 及其 reference 中的两篇文章）和代码（https://github.com/TL-System/ns.py）了解 WFQ的算法过程和ns.py代码实现。
2. 建议 2 条线并行：
   1. 代码：使用 ns.py 的 WFQ 代码改成 regenerative method 的形式（参考我们 repo 的default version），估计每个队列响应时间的期望及其CI。
   2. 理论：分析系统在每个时刻的 likelihood ratio（参考我们priority 文章中的定义），并定义一个合适的停时。具体来说，每个时刻的 likelihood ratio (LR) 取决于我们用到哪些 random variables，有的用到了具体的时刻，LR就是 density 之比；有的只是需要直到目前没发生，LR就是 tail probability的比例。停时应该选到我们知道这个 cycle 存在一个响应时间大于 gamma 的时间，例如优先级队列我们就用到了某个包到达的时候的目前知道的响应时间。  
      所以流程应该是：通过分析算法过程，确定一个合适的停时（如某个包到达的时候的目前知道的响应时间），由此知道哪些随机变量需要或不需要 realized，之后通过需要 realize的随机变量来计算 LR。
3. IS+CE：把 2b 加到 2a 得到 IS 的 WFQ。再代入 CE 的公式，可以迭代算出CE参数。

Experiments:

Lambda1, lambda2 = 0.1, 0.2 mu1=mu2=1

1. 2a)中的 mean 及其 CI。
2. 3 中的 CE参数下的 IS，考虑gamma 等于 5 和 10 下的 两个队列的 tail probability。