

## 1. 실험 목적

몬테카를로 시뮬레이션은 난수를 활용해서  $\pi$  값을 추정하는 방법입니다.

이번 실험에서는 스레드 수(1, 2, 4, 8)에 따라 실행 시간이 어떻게 달라지는지 알아보고,

데이터 크기가 결과에 어떤 영향을 미치는지 분석했습니다.

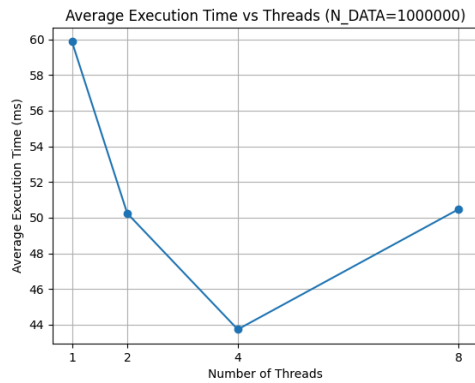
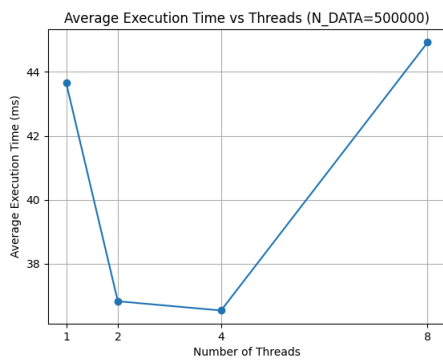
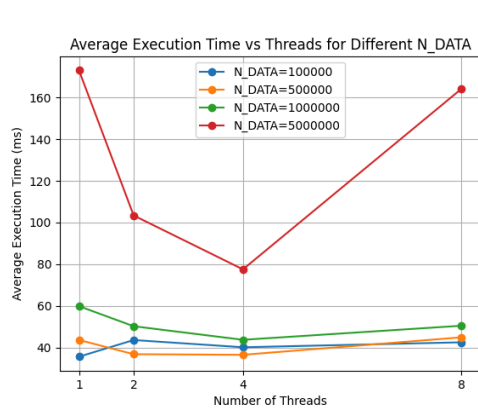
---

## 2. 실험 방법

- 실험 환경:
    1. pthread 라이브러리를 이용해 병렬 처리 구현
    2. pthread\_mutex 사용해 thread safe하게 total\_count를 업데이트하도록 구현
  - 실험 변수:
    1. 스레드 수: 1, 2, 4, 8
    2. NDATA: 100,000, 500,000, 1,000,000, 5,000,000
  - 실험 과정:
    1. 각 NDATA와 스레드 수 조합에 대해 프로그램 실행
    2. 실행 시간을 측정하고 평균값 계산
    3. 결과를 그래프로 시각화
-

### 3. 실험 결과

#### (1) 그래프



#### (2) 그래프 분석

##### 1. NDATA=100,000:

- 스레드 수가 많아질수록 실행 시간의 감소폭이 작았습니다.
- 8개 스레드에서는 실행 시간이 오히려 증가했는데, 이는 워크로드가 작아 스레드 관리에 드는 비용이 더 컸기 때문으로 보입니다.

##### 2. NDATA=500,000:

- 스레드 수가 2개나 4개일 때 실행 시간이 눈에 띄게 감소했습니다.

- 그러나 8개 스레드에서는 뮤텝스 경쟁으로 인해 성능이 저하되었습니다.
- 3. **NDATA=1,000,000**
  - 데이터 크기가 커지면서 병렬 처리 효과가 더 명확히 나타났습니다.
  - 4개 스레드에서 최적의 실행 시간을 기록했습니다.
- 4. **NDATA=5,000,000**
  - 가장 큰 데이터 크기에서 병렬 처리가 가장 효과적이었습니다.
  - 하지만 8개 스레드에서는 스레드 관리 비용과 뮤텝스 경쟁이 높아져 시간이 더 걸리는 모습을 보였습니다.

### (3) 종합 그래프 분석

- NDATA가 클수록 스레드 수가 4개일 때 성능이 가장 좋았습니다.
  - 8개 스레드에서는 모든 NDATA에서 실행 시간이 증가하는 경향이 있었습니다.
- 

## 4. 문제 원인

1. 스레드 관리 비용:
    - 스레드 수가 많아질수록 생성과 스케줄링에 드는 비용이 증가했습니다.
  2. 뮤텝스 경쟁:
    - `pthread_mutex`로 `total_count`를 업데이트할 때 여러 스레드가 동시에 접근하려 하면서 병목 현상이 발생했습니다.
  3. 작은 데이터 크기:
    - NDATA가 작을 경우 병렬 처리로 얻는 이정보다 관리 오버헤드가 더 크게 작용했습니다.
-