ASC명령으로 정렬한 경우

시간은 100번의 시행의 평균 (단위: 초)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 종류  데이터 | insertion  sort | selection  sort | bubble  sort | quick  sort | merge  sort | heap  sort | python  sort |
| 1000 | 0.00014 | 0.06400 | 0.27723 | 0.09095 | 0.00377 | 0.01057 | 0.00000 |
| 2000 | 0.00102 | 0.24589 | 1.13449 | 0.30139 | 0.00758 | 0.02509 | 0.00000 |
| 3000 | 0.00134 | 0.53321 | 2.32621 | 0.70297 | 0.01287 | 0.04278 | 0.00000 |
| 4000 | 0.00151 | 1.0107 | 4.50329 | 1.25955 | 0.01598 | 0.05285 | 0.00000 |
| 5000 | 0.00200 | 1.58625 | 7.15614 | 2.03296 | 0.01972 | 0.06709 | 0.00000 |

python\_sort와 insertion\_sort가 매우 빠른 처리를 보여줌을 알 수 있었습니다. bubble\_sort는 데이터의 양이 많아질수록 속도에서 불리해지는 경향을 보임을 알 수 있었습니다.

DESC명령으로 정렬한 경우

시간은 100번의 시행의 평균 (단위: 초)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 종류  데이터 | insertion  sort | selection  sort | bubble  sort | quick  sort | merge  sort | heap  sort | python  sort |
| 1000 | 0.39216 | 0.05059 | 0.25601 | 0.06034 | 0.00360 | 0.00951 | 0.00000 |
| 2000 | 1.62897 | 0.20611 | 1.07016 | 0.22163 | 0.00617 | 0.02085 | 0.00000 |
| 3000 | 3.72284 | 0.43494 | 2.23132 | 0.49954 | 0.01039 | 0.03228 | 0.00000 |
| 4000 | 6.94151 | 0.76003 | 3.51465 | 0.74914 | 0.01254 | 0.04051 | 0.00000 |
| 5000 | 11.84612 | 1.02510 | 7.58736 | 0.81830 | 0.01410 | 0.04280 | 0.00000 |

ASC명령과는 다르게 insertion\_sort의 실행 속도가 눈에 띄게 느린 것을 발견할 수 있었습니다. ASC명령과 마찬가지로 bubble\_sort는 불리함을 알 수 있었습니다.

quick\_sort와 merge\_sort에는 정렬이 반대로 되는 오류가 있음을 발견하였습니다.

RANDOM명령으로 정렬한 경우

시간은 10번의 시행의 평균 (단위: 초)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 종류  데이터 | insertion  sort | selection  sort | bubble  sort | quick  sort | merge  sort | heap  sort | python  sort |
| 1000 | 0.20610 | 0.04470 | 0.20400 | 0.00200 | 0.00360 | 0.00770 | 0.00000 |
| 2000 | 0.94570 | 0.16440 | 0.79340 | 0.00400 | 0.00830 | 0.01680 | 0.00000 |
| 3000 | 1.84219 | 0.36900 | 1.81920 | 0.00690 | 0.01199 | 0.02530 | 0.00000 |
| 4000 | 3.49799 | 0.66360 | 3.21970 | 0.00919 | 0.01720 | 0.03530 | 0.00000 |
| 5000 | 5.37919 | 1.12470 | 5.21040 | 0.01199 | 0.02110 | 0.04689 | 0.00000 |

위의 명령들보다는 아니지만 insertion\_sort와 bubble\_sort가 상대적으로 속도에서 불리한 면이 있었습니다.

quick\_sort와 merge\_sort는 DESC방식과 마찬가지로 오류가 발생하였는데, 리스트 내의 데이터들이 무질서하게 배치가 됨을 알게 되었습니다. sort함수가 작성될 때 recursion부분에서 문제가 생겼지 않을까 합니다.

세 명령에 대한 시간을 계산할 때 컴퓨터의 상태가 차이가 컸기 때문에 신뢰성 있는 정보라고 생각하지는 않습니다.