

**알고리즘 레포트**

**Msd-radix sort**



|  |  |
| --- | --- |
| **분 반 :** | **2반** |
| **제출 일자 :** | **2018.05.23** |
| **담당 교수 :** | **백정엽** |
| **학 과 :** | **컴퓨터공학** |
| **학 번 :** | **20146110** |
| **이 름 :** | **문태현** |

1. 실행결과값 (표)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | selection | insertion | quick | merge | msd-radix |
| 1000 | 1.2486 | 0.66451 | 0.089 | 0.1299 | 0.075 |
| 10000 | 119.462 | 63.8489 | 1.149 | 1.6879 | 0.421 |
| 50000 | 2973.967 | 1597.647 | 6.691 | 9.543 | 2.391 |
| 100000 | 11897.14 | 6709.925 | 14.108 | 20.1023 | 5.596 |
| 146110 | 26104.04 | 14326.34 | 21.418 | 30.2413 | 9.259 |
| 200000 | 47579.23 | 26794.42 | 29.607 | 42.3132 | 13.98 |
| 500000 | 298370.3 | 168177.8 | 79.484 | 113.7822 | 25.964 |
| 750000 | 675084.7 | 381153.3 | 123 | 173.9852 | 34.775 |
| 1000000 | 1252619 | 686662.7 | 168.94 | 238.9344 | 43.105 |

2. 그래프

Selection, insertion, quick, merge, msd-radix sort를 비교하는 그래프이다.

Quick,merge,msd-radix sort 는 너무 빨라서 보이지 않는다.

Log-log scale 그래프이다.

Selection<insertion<merge<quick<msd-radix 순으로 속도가 나타난다.

3. 결론

Selection과 insertion은 O(N\*N)의 시간복잡도를 나타내는 걸로 알려져있고 다른 sorting에 비해 많이 느린 것을 확인할 수 있다. Merge와 quick sort는 O(NlogN)으로 알려져있고 속도가 아주 빠르다. 그보다 빠른 msd\_radix\_sort는 적용할 수 있는 범위는 한정되지만 우리가 실험한 input인 unsigned integer에서는 merge나 quick보다 훨씬 빠른 속도를 보이고 있다. 이론상으로 O(N)의 시간 복잡도를 가지는 데 log-log scale그래프를 확인해보면 어느정도 확연히 차이가 남을 확인 할 수 있다.

그런데 순수하게 msd\_radix\_sort 로만 구현하게되면 input의 개수가 작을 때 (약 십만아래) quick이나 merge보다 느린 sorting속도를 보인다. 이는 엄청난 양의 small subarray를 처리해야 하기 때문으로 보이고(많은 recursive) subarray에서 count배열을 초기화 하는 작업이 지대한 영향을 미치는 것으로 보인다. 이를 해결하기 위해서

<http://www.informit.com/articles/article.aspx?p=2180073&seqNum=3>

이문건을 참조하였는데 일정크기 이하의 subarray에서는 insertion sort를 사용 하는 것이다. 이렇게하면 그래프에 보이는 것과 같이 input이 작을 때도 다른 sort보다 속도가 빠르다.