
Insertion & merge sort

2016 Algorithm class homework

00반 201102513 조상경

목차

1. 구현

2. 결과 및 분석

(A) Performance overview

(B) Insertion & Binary insertion

(C) Merge & 3-way merge

1. 구현

사용 언어 : C

사용된 컴파일러/운영체제 : Clang - 703.0.31 / Mac OSX 10.11.6

a) Insertion sort

강의 자료의 pseudo코드를 c로 구현

b) Binary-insertion sort

Binary search를 구현 한 후에, 이를 insertion sort에 접목.

기존에 insertion search에서 정렬이 되어있는 왼쪽의 구역에 대해 값을 넣기 위해 순차적으로 찾아 들어가기 때문에 이를 binary search를 통해 위치를 찾는 방식으로 구현함.

처음에는 위치를 찾고, 나머지 값들을 옮기기 위해 while문을 똑같이 쓰면 오히려 코드 시행 횟수에서 손해를 보기 때문에 memmove를 이용하여 값들을 밀어냄.

c) Merge sort

배열을 받아서 두 개의 새로운 배열로 나눈 후, 이를 재귀적으로 호출하여 merge로 정렬 후 합쳐나감.

d) 3way merge sort

기존의 merge sort에서 두 개로 나눴던 배열을 세 개로 확장하여 재귀적으로 호출함.

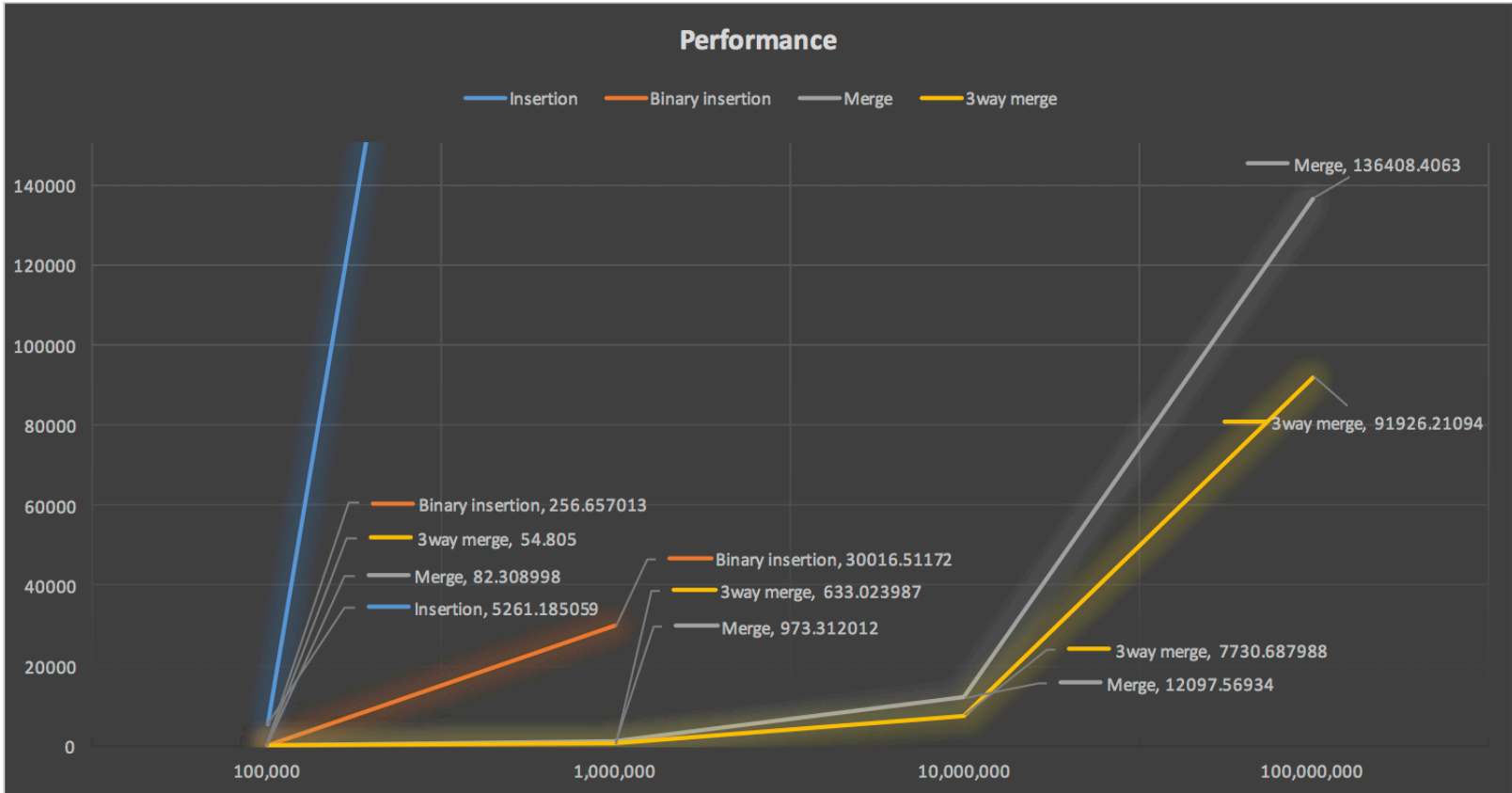
최종 조각의 길이가 2일때는 머지소트를 부르는 대신에 두 값을 스왑하여 이론적 성능에 근접하도록 함

2. 결과 및 분석

(A) Performance overview

(단위 : millisecond)

	100,000	1,000,000	10,000,000	100,000,000
Insertion	5261.185059	517952.9375	-	-
Binary insertion	256.657013	30016.51172	-	-
Merge	82.308998	973.312012	12097.56934	136408.4063
3way merge	54.805	633.023987	7730.687988	91926.21094



데이터의 개수가 적을 때는 각 알고리즘간의 성능차가 크게 나타나지 않았지만, 데이터의 개수가 점점 커질수록, 성능차가 크게 벌어지는 것을 확인할 수 있었다. 성능은 순서대로 3way merge, merge, binary insertion, insertion 순으로 높게 나타나는 것을 확인하였다.

(B) Insertion vs Binary insertion

<search 횟수>

	50	100,000	1,000,000
Insertion	541	2405081002	240377688628
Binary insertion	271	1622840	19548111

Insertion 과 Binary insertion의 차이를 비교하기 위해, 각각의 함수의 루프 영역의 통과 횟수를 체크하여 보았다. Insertion은 key값이 들어갈 곳을 찾기 위해 정렬된 앞쪽의 배열을 순차적으로 찾아가게 되는데, 이론적으로 binary search를 이용하면 key값이 들어갈 곳을 찾기 위해 걸리는 시간이 더 적어야 한다. 실제 데이터 상으로도, binary insertion이 값을 찾기 위해 들어간 탐색횟수가 더 적은 것을 확인할 수 있었다.

(C) Merge vs 3-Way merge

<Merge() 호출 횟수>

	100,000	1,000,000	10,000,000	100,000,000
Merge	99999	999999	9999999	99999999
3Way-merge	29524	265720	2825546	35429918

Merge와 3-way merge에서 각각 merge()함수의 호출 횟수를 체크하였다. 3-way merge의 merge함수의 호출 횟수가 Merge의 merge함수 호출 횟수보다 적어진 것을 확인할 수 있었다.