

Mergesort 비틀어 보기

Mergesort에서 둘로 나누는 대신 16 개로 나누어도 sorting은 된다. 즉, 최상위 레벨에서 mergesort를 16 번 부른 다음 merge를 한다. Merge는 2개의 정렬된 배열을 하나의 정렬된 배열로 만드는 대신, 최대 16개의 정렬된 배열을 하나의 정렬된 배열로 만든다.

이렇게 하면 mergesort의 recursive call의 깊이가 $\sim \log_2 n$ 에서 $\sim \log_{16} n$ 로 떨어져 시간이 절약된다. 반면 2개 대신 16개를 merge하는 부담으로 constant factor 값이 커진다.

프로그래밍 연습의 목적으로 여러분은 3가지 mergesort를 구현해본다.

1. Original mergesort
2. 16개짜리 mergesort를 하되, 16개 중의 최솟값을 찾아 내리기 위해 모두 비교해서 최솟값을 찾는 방식으로 구현
3. 16개짜리 mergesort를 하되, 16개 중의 최솟값을 찾아 내리기 위해 16개짜리 minheap을 만들어 구현. 이 경우 minheap은 전형적인 배열에 연속적으로 값이 들어있는 방식으로 할 수 없으므로 다른 방법으로 구현해야 함.

위 구현 중 3에 대해서는 여러분이 minheap을 구현한 방식을 간략하게 한 페이지 이내로 기술해서 제출할 것.

주어지는 입력에 대해 위 3개의 mergesort를 구현한 다음, 각 버전의 mergesort에 대해 결과가 맞다는 것을 간단히 확인하고 수행 시간을 출력하도록 한다.

<제한 조건>

제한시간: 총 10문제 합하여 180초

입력 제한: 옳게 작동한다는 것을 확인하기 쉽도록 모든 원소는 서로 다른 양의 정수 값을 갖는다. 입력 원소의 총 수는 10개에서 100만개 사이다. 각 원소는 32 bit 이내로 표현할 수 있는 크기의 양의 정수다. 50만개 이상의 원소를 갖는 입력 수는 5개다.

[입력]

입력 파일에는 10개의 테스트 케이스가 주어진다. 각 케이스는 두 줄로 이루어진다. 첫 줄에는 입력 원소의 총 수를 나타내는 정수 N 이 주어지고, 둘째 줄에는 N 개 원소가 나열된다. 원소들 사이의 구분은 공백으로 한다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 "#x" 의 형식으로 출력한 후(여기서 x는 테스트 케이스 번호), 공백을 하나 둔 다음 아래 항목들을 차례로 기록한다. 출력 결과물을 "output1.txt"로 저장한다.

제공된 answer1.txt에는 문제의 답만 기록되어 있으나, 제출 시에는 아래의 출력예시에 맞추어 소요되는 시간까지 출력해야 한다.

- a. 정렬된 결과 4의 배수번째 원소들을 7로 나누었을 때의 나머지를 모두 더한 값

(단, 배열은 0번째 부터 시작한다고 가정한다. 0번째, 4번째, 8번째 ... 4n번째 해당)
b. 본문에서 기술한 버전 1, 2, 3의 mergesort 각각을 수행하는데 소요되는 시간

즉, 출력은 1개의 정수값과 3개의 실수값으로 구성된다.

[입출력 예]

입력

10	← 1번 케이스의 시작줄
10 12 2 1 15 3 6 5 14 7	
10	← 2번 케이스의 시작줄
29 6 12 1 9 17 18 24 4 30	
...	

출력

#1 7 0.001 0.002 0.004	// 시간은 단순 예시
#2 7 0.001 0.002 0.004	// 시간은 단순 예시
...	