CS325 Winter 2013: Implementation 1

Daniel Reichert
Trevor Bramwell

February 5, 2013

Asymptotic Analysis

1. In algorithm 1 the problem is not branched into any subproblems. With the nested for loops there is a comparison made between every element in the input, leading to an intuitive asymptotic complexity of $O(n^2)$

2. In algorithm 2 the problem is branched into 2 subproblems for size $n/2$ at each level. It is similar to algorithm 3, but the counting of inversions is done inefficiently. This is represented in the master theorem as the cost of combining the subproblems. Because of the naive nature of the merge and count approach, the additional cost of combining ends of being the dominating term in the master theorem as it is greater than $nlogn$. Every time that two lists are merged, each element is compared to every other element. Since there are $log_2 n$ merges that take place and each merge compares itself to every other item in the list this is a $O(n^2 logn)$ algorithm.

3. In algorithm 3 the problem is branched into 2 subproblems of size $n/2$ at each level. The depth of the problem is $log_2 n$ and the width is $n^{log_2 2}$. From the master theorem we know that A/B^D determines the run time complexity. Since this case is $2/2^1 = 1$, we know that the Asymptotic complexity is $O(nlogn)$.

Testing

verify.txt	test_input.txt
9670	249310
10567	252709
9282	253719
9269	249315
9675	247789
10378	254833
9911	239844
9790	257527
9580	241669
9965	255628

Extrapolation and Interpretation
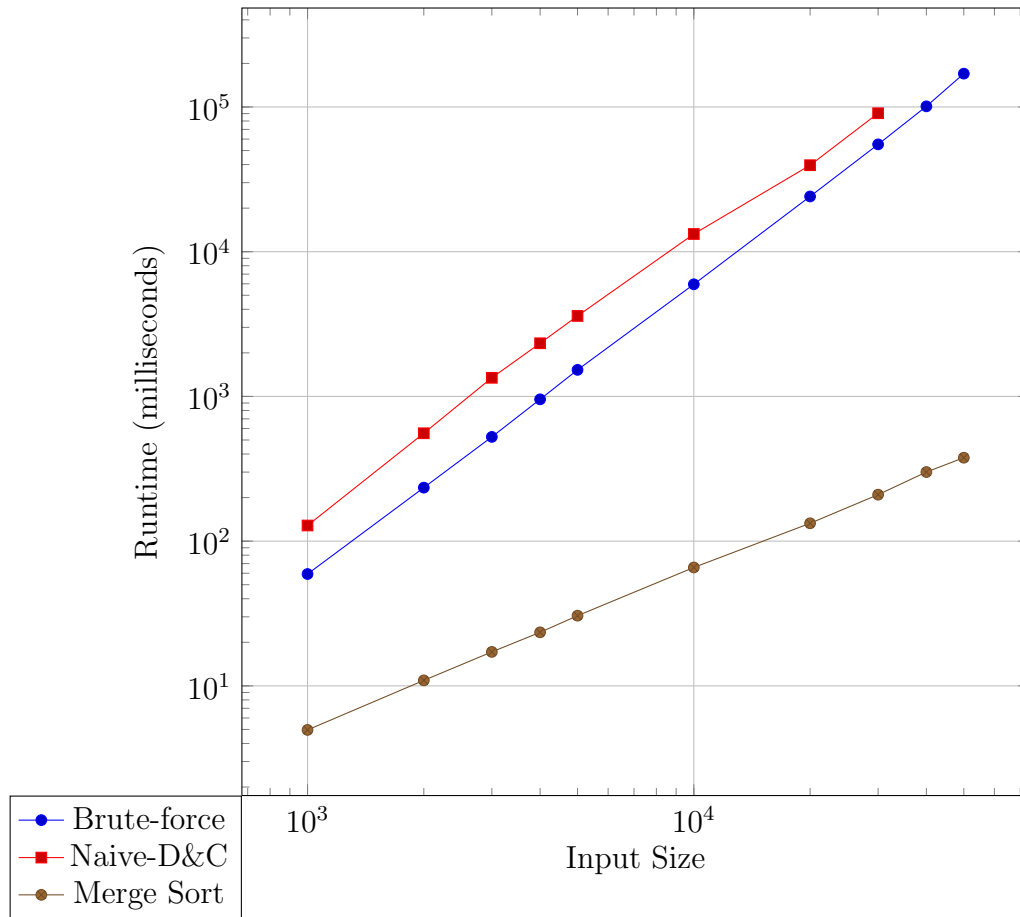
Slope of Lines

- Brute-force: 2

- Naive-D&C: 2

- Merge Sort: 1