15-451 Assignment 01

Karan Sikka
ksikka@cmu.edu

September 1, 2014

1a.

Compute $n^3/4$ in constant time.
Use DeterministicSelect to select the $n^{3/4}$th largest number in $O(n)$ time.
Then filter out the elements greater than or equal to it in $O(n)$ time.
Now sort the $n^{3/4}$ numbers using mergesort in $O(n^{3/4}log(n^{3/4}))$ time.
The algorithm seems to be dominated by the latter expression, but it can be reduced to $O(n)$ as follows:

$O(n^{3/4}log(n^{3/4}))$
$\leq O(\frac{3}{4}n^{3/4}log(n))$

Notice that $O(n^{1/4}) \geq O(log(n))$ so we can make the following substitution:

$\leq O(\frac{3}{4}n^{3/4}n^{1/4})$
$\leq O(n)$

1b.

Pair up the elements, and for each pair, compare the elements. ($\frac{n}{2}$ comparisons)
Call the larger element a "winner" and the smaller element a "loser".
Among the $\frac{n}{2}$ winners, find the max by going one by one keeping track of the max so far. The minus one term is because you don't have anything to compare the first element to - you assume it as the max at first. ($\frac{n}{2} - 1$ comparisons)
This is the max of all the elements.

Among the $\frac{n}{2}$ losers, find the min by going one by one keeping track of the min so far. ($\frac{n}{2} - 1$ comparisons)

The sum of all the comparisons is $(\frac{n}{2}) + (\frac{n}{2} - 1) + (\frac{n}{2} - 1) = \frac{3n}{2} - 2$

2

Lorem ipsum