Aviad Hahami
302188347

# Data Structures
## Ex. 05

1. Given data:

$n_1 = 500K$

$n_2 = 1M$

$f(n) \approx d * c * nlog(n)$

a)

$f(n_1) / f(n_2) \approx \frac{d*c*(500K)log(500K)}{d*c*(1M)log(1M)} \approx \frac{(500K)log(500K)}{(1M)log(1M)} \approx \frac{1}{2} * 0.949828334 \approx 0.474914167$

b)

0.64398127023

c)

0.35599507232 of an error

2.

a)This calculation is similar to the one in 1.a, so I will just calculate the $\frac{1}{2} * \frac{(500K)log_3(500K)}{(1M)log_3(1M)}$

$\frac{1}{2} * \frac{log_3(500K)}{log_3(1M)} \approx \frac{1}{2} * 0.94983 \approx 0.474915$

b)

0.242048855

c)

0.96206257616

3.

$f(n) \approx d * c * n$

a) $\approx \frac{1}{2}$

b)

0.56573868634

c)

0.13147737268

4.

$f(n) \approx d * c * n$

a) $\approx \frac{1}{2}$

b)

0.90309224021

c)

0.80618448042

# Data Structures
## Ex. 05

5.

No.

Assuming we “hit” merge-sort's worst case (maximum comparisons) we might get a result worse than an other array. Another thing we need to put in mind is the aspect of OS demands as the calculating process “fights” over the CPU/IO components just like any other system process. Therefore if we have “Defragment” running on our OS, the sorting results will be slower than running the sort without it in the background.