Problem session 1

Dibran Dokter 1047390

September 11, 2019

1

1.1

$10^6 = 1\ million,\ 10^6 * 60 = 60\ million$
a) $n = \sqrt{60\ million}$
b) $n =\approx 3950500$
c) $n = 15.556$
d) $n =\approx 153300$
e) $n =\approx 1.1689$
f) $n =\approx 12.92$
g) $n = 60.000.000$

1.2

a) $f = \Theta(g)$
b) $f = \Theta(g)$
c) $f = \Theta(g)$
d) $f = \Omega(g)$
e) $f = \mathcal{O}(g)$
f) $f = \mathcal{O}(g)$
g) $f = \Theta(g)$
h) $f = \mathcal{O}(g)$

1.3

$n \in \mathcal{O}(2^n)$
$N_0 = 1,\ c\ =\ 1$
$f(1) \leq 1 \cdot g(1)\ for\ all\ n \geq N_0$
$f(1)\ =\ 1, g(1)\ =\ 1$
$f(2)\ =\ 2, g(2)\ =\ 4$
$thus\ for\ all\ n\ > N_0,\ g(n)\ \geq\ f(n)$
$f(n) \leq c \cdot g(n)$
$f(n) = \mathcal{O}(g)$

1.4

a) The worst case is $\mathcal{O}(n)$, if the element to find is not in the array.

b) The best case is $\mathcal{O}(1)$, if the element to find is the first element in the array.

1.5

a) The worst case is $\mathcal{O}(n^2)$.

b) The best case is $\mathcal{O}(1)$, when there is only one element in the array.

When there are more elements the best case is $\mathcal{O}(n^2)$.

c) There is no way to improve an algorithm with complexity $\mathcal{O}(1)$.

We can improve the complexity of the case $\mathcal{O}(n^2)$ by first checking whether the array is already sorted in a for loop. This would give a complexity of $\mathcal{O}(n)$.