Name:

Mohammed maher hassan

Student Number:

120170672

Assignment 5

Prof. Mohammed Alhanjouri

3.1 Algorithms

Q4.

procedure maxdiff($a_1, a_2, ...a_n$: integers with $n \geq 2$)
maxdiff:=$a_2 - a_1$
for i:=3 to n
    diff:=$a_i - a_{i-1}$
    if maxdiff>diff **then** maxdiff:=diff
return maxdiff

Q8.

We call the algorithm "lasteven2" and the input is a list of n integers.

procedure lasteven2($a_1, a_2, ...a_n$: integers with $n \geq 1$)

We initially define the variable k as 0 (k will determine the last even integer).

k:=0

For every integer between 1 and n, if a_i is even (divisible by 2), then we replace the current value of k by i (since i represents the location of the even integer).

for i:=1 to n
    if a_i is even **then** k:=i

Finally we return the variable k, which at the end of the for-loop will contain the last even integer in the list.

return k

Q16

We call the algorithm "minimum" and a list of natural numbers $a_1, a_2, ..., a_n$

procedure minimum($a_1, a_2, ...a_n$: natural numbers with $n \geq 1$)

We initially define the minimum as the first element in the list (if it is not the minimum, then this value will be adjusted later in the algorithm).

min:=a_1

For the 2nd to nth element in the list, we then compare it with the current minimum in that step. If the value is smaller, then we reassign the value to the minimum.

for $i := 2$ to n
  if a_i <min **then** min:=a_i

Finally we return the found minimum of the list.

return min 0

Q38.

First pass

d f k m a b, d f k m a b , d f k m a b , d f k a m b , d f k a b m

second pass

d f k a b m, d f k a m b , d f a k b m , d f a b k m

third pass

d f a b k m, d a f b k m , d a b f k m

Forth pass

a d b f k m , a b d f k m

3.2 The Growth of Functions

Q2.

(a) $O(x^2)$
(b) $O(x^2)$
(c) $O(x^2)$
(d) Not $O(x^2)$
(e) Not $O(x^2)$
(f) $O(x^2)$

Q8.

(a) $n = 4$
(b) $n = 5$
(c) $n = 0$
(d) $n = 0$

Q22.

$(\log n)^3 < \sqrt{n}\log n < n^{99} + n^{98} < n^{100} < (1.5)^n < 10^n < (n!)^2$

Q26.

(a) $O(n^3 \log n)$
(b) $O(6^n)$
(c) $O(n^n n!)$

3.3 Complexity of Algorithms

Q2.

$O(n^2)$

Q12.

a) There are three loops, each nested inside the next. The outer loop is executed n times, the middle loop

is executed at most n times, and the inner loop is executed at most n times. Therefore the number of times

the one statement inside the inner loop is executed is at most n3 . This statement requires one comparison,

so the total number of comparisons is O(n3).

(b) The number of comparisons is $n^3 - 2n^2 + n$, while $n^3 - 2n^2 + n$ is $\Omega(n^3)$.

Since the number of comparisons is $O(n^3)$ and $\Omega(n^3)$, the number of comparisons is also $\Theta(n^3)$.

Q18.

(a)

$$n = 10$$

The time that the algorithm takes is the product of the time per operation and the number of operations:

$$\begin{aligned} T &= 10^{-9}(2n^2 + 2^n) \text{ seconds} \\ &= 10^{-9}(2(10^2) + 2^{10}) \text{ seconds} \\ &= 1224 \times 10^{-9} \text{ seconds} \\ &= 1224 \text{ nanoseconds} \end{aligned}$$

(b)

$$n = 20$$

The time that the algorithm takes is the product of the time per operation and the number of operations:

$$\begin{aligned} T &= 10^{-9}(2n^2 + 2^n) \text{ seconds} \\ &= 10^{-9}(2(20^2) + 2^{20}) \text{ seconds} \\ &= 1049376 \times 10^{-9} \text{ seconds} \\ &= 0.001049376 \text{ seconds} \\ &= 1,049,376 \text{ nanoseconds} \end{aligned}$$

(c)

$$n = 50$$

The time that the algorithm takes is the product of the time per operation and the number of operations:

$$\begin{aligned} T &= 10^{-9}(2n^2 + 2^n) \text{ seconds} \\ &= 10^{-9}(2(50^2) + 2^{50}) \text{ seconds} \\ &= 1,125,899,906,847,624 \times 10^{-9} \text{ seconds} \\ &\approx 1,125,900 \text{ seconds} \cdot \frac{1 \text{ minute}}{60 \text{ seconds}} \\ &= 18765 \text{ minutes} \cdot \frac{1 \text{ hour}}{60 \text{ minutes}} \\ &\approx 313 \text{ hours} \cdot \frac{1 \text{ day}}{24 \text{ hours}} \\ &\approx 13 \text{ days} \end{aligned}$$

(d)

$$n = 100$$

The time that the algorithm takes is the product of the time per operation and the number of operations:

$$\begin{aligned}
T &= 10^{-9}(2n^2 + 2^n) \text{ seconds} \\
&= 10^{-9}(2(100^2) + 2^{100}) \text{ seconds} \\
&= 1{,}267{,}650{,}600{,}228{,}229{,}401{,}496{,}703{,}225{,}376 \times 10^{-9} \text{ seconds} \\
&\approx 1.27 \times 10^{21} \text{ seconds} \cdot \frac{1 \text{ minute}}{60 \text{ seconds}} \\
&\approx 2.11 \times 10^{19} \text{ minutes} \cdot \frac{1 \text{ hour}}{60 \text{ minutes}} \\
&\approx 3.52 \times 10^{17} \text{ hours} \cdot \frac{1 \text{ day}}{24 \text{ hours}} \\
&\approx 1.47 \times 10^{16} \text{ days} \cdot \frac{1 \text{ year}}{365 \text{ days}} \\
&\approx 4.02 \times 10^{13} \text{ years}
\end{aligned}$$

Thus about 40.2 trillion years.