Faculty of computer and information
Department of computer science

MINIA UNIVERSITY

Algorithm analysis & design

1) Prove that:

- $\frac{1}{2}\, n\,(n-1) = \Theta(n^2)$
- $15n^3 + n^2 + 4 = O(n^3)$
- $n^3 + 4n^2 = \Omega(n^2)$
- $n^2/2 - 3n = \Theta(n^2)$
- $2^{n+1} = O(2^n)$
- $n^{3=}\ \Omega(n^2)$

2) Order the following functions by growth rate (fastest is first) (all log base 2)

1) $f(n) = n^{\log 7}$ 2) $f(n) = \sqrt{e^{\log n} + n^3}$ 3) $f(n) = \sum_{i=1}^{n} i$

4) $f(n) = \log^k n$ 5) $f(n) = n^3$ 6) $f(n) = \sqrt[3]{7^{\log n}}$

7) $f(n) = \mathrm{n}$ 8) $f(n) = 4^{\log n}$ 9) $\frac{1}{\sqrt{\log n}}$

10) $f(n) = \mathrm{n} \times \log(\log(\mathrm{n}))$

3) Write the big-Θ expression (tight bound) to describe the number of operations required for the following C++ code:

```
1) for (i = 2 ; i < N ; i *= i)
                A[i] = i ;
2) Sum = 0
        for (j = 0 ; j <= N ; j++)
          for (k = 0 ; k <= j × j ; k++)
               if (k % j == 0)
                    for (m = 0 ; m <= k ; m++)
                         Sum++;
```

Note:

- Delivery of Assignment (1)
    - ❖ **Due date:** MO 23-NOV

By: *Eng. Asmaa Hamad*