CSC148 Runtime Analysis

Michael Miljanovic

Exercise 1

For each function f in the left column of the following table, choose one expression $\mathcal{O}(g)$ from the following list:

$\mathcal{O}(\frac{1}{n})$, $\mathcal{O}(1)$, $\mathcal{O}(\log_2 n)$, $\mathcal{O}(n)$, $\mathcal{O}(n \log_2 n)$, $\mathcal{O}(n^2)$, $\mathcal{O}(n^{10})$, $\mathcal{O}(2^n)$, $\mathcal{O}(10^n)$, $\mathcal{O}(n^n)$

such that $f \in \mathcal{O}(g)$. Use each expression only once.

f	$\mathcal{O}(g)$
$3 \cdot 2^n$	
$\frac{2n^4+1}{n^3+2n-1}$	
$(n^5+7)(n^5-7)$	
$\frac{n^4-n\log_2 n}{n^2+1}$	
$\frac{n\log_2 n}{n-5}$	
$8+\frac{1}{n^2}$	
2^{3n+1}	
$n!$	
$\frac{5\log_2 n+1}{1+n\log_2 3n}$	
$(n-1)\log_2(n^3+4)$	

Exercise 2

Determine the number of steps taken by the following algorithm.

```
from math import log2, floor

def f(n):
  lst = []
  for i in range(floor(log2(n))):
    lst.insert(0, 0)
  return lst
```

Exercise 3

Determine the number of steps taken by the following algorithm.

```
def f2(n):
    i = 4
    while i < n:
        j = 1
        while j < n:
            j = j * 3
        k = 0
        while k < n:
            k = k + 2
        i = i + 1
```