Homework 1 : Notation

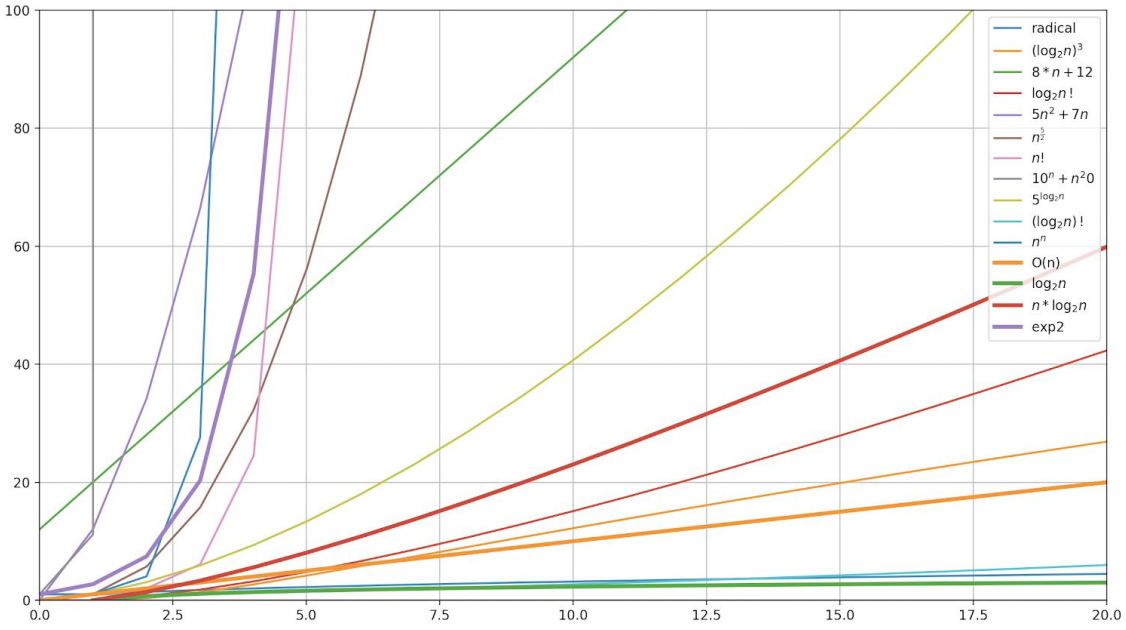
Solution:

1. quick way to compare growth of two given functions f(x) and g(x) :

   $\lim_{x\to\infty} \frac{f(x)}{g(x)} = if\ 0 \leftrightarrow growth\ of\ f(x)\ is\ less\ than\ g(x)\ ,\ \infty \leftrightarrow\ growth\ of\ f(x)$
   $is\ more\ than\ g(x),\ k\ \neq 0 \leftrightarrow they\ both\ grow\ equally$

   - $\lim_{n\to\infty} \frac{log(n)}{n} = 0 \rightarrow log(n)\ \varepsilon\ O(n)$
   - $\lim_{n\to\infty} \frac{n}{nlog(n)} = 0 \rightarrow n\ \varepsilon\ O(nlog(n))$
   - $nlog(n) \leq n^2\ N = 0\ ,\ c = 1,\ n \geq 0$
   - $2^n \geq 5^{ln(n)}\ \ N = 0\ ,\ c = 1\ ,\ n \geq 0$

2. $logn < (logn)^3 < \sqrt{n} < 8n + 12 < (logn)! < log(n!) < n.logn < 5n^2 + 7n$
   $< n^{\frac{5}{2}} < n^3 < 5^{logn} < e^n 10n + n^{20} < n! < n^n < n^n + ln(n)$



3. The first loop is going from 0 to n , so repetition of "i<n" will be n times . second loop is going to execute for $2^k = n$, so the "j>1" will be executed $logn$ times. In the end the hole code is going to run in $O(nlog(n))$ order .
4. Answers:

a. 25+32 = 57
b. Each loop is going to execute n times and they're not nested so the function runs in O(n) order .
c. We can do the k's multiplication in the first loop .

5.

exercise > r.py > ...

```python
minimum = 1e7
def rec(arr):
    print(arr)
    if len(arr) == 1:
        return [arr[0]]

    mid = int(len(arr)/2)
    left = rec(arr[:mid])
    right = rec(arr[mid:])

    if isinstance(left, list):
        left_sum = sum(left)
    else:
        left_sum = left

    if isinstance(right, list):
        right_sum = sum(right)
    else:
        right_sum = right

    global minimum
    minimum = min(minimum, abs(left_sum - right_sum))

    print("min is: ", minimum)
    return [*left, *right]


print(rec([1, 4, 4, 6, 4, 5, 7, 8]))
print("minimum is: ", minimum)
```

TERMINAL SQL CONSOLE: MESSAGES DEBUG CONSOLE PROBLEMS OUTPUT

```
[4]
[6]
min is:  2
min is:  2
[4, 5, 7, 8]
[4, 5]
[4]
[5]
min is:  1
[7, 8]
[7]
[8]
min is:  1
min is:  1
min is:  1
[1, 4, 4, 6, 4, 5, 7, 8]
minimum is:  1
(env) → exercise
```

6. The first computer will solve the problem of n=1000 size in one minute . the new computer will solve n=1000 size problem in $\frac{1}{1000}$ minute , so it can solve a problem of size n = 10^6 in one minute .
   a) T(n) ∈ $\theta(n)$ n=1000 n = 1000*1000 = 1000000 n* = 10^6
   b) T(n) ∈ $\theta(n^3)$ n=10 n = 10*10*10= 1000*1000 n* = 10^6
   c) T(n) ∈ $\theta(10^n)$ n=3 10^n = 10^3 * 1000 n*=10^6