DS HW4

Nicholas Lee and Howie Li

December 24, 2022

Q1

{{a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b,c}}

Q2

115975

Q3

Assume A = {a,b}
$A^2 = \{(a,a),(a,b),(b,a),(b,b)\}$
$A^2 \times A = \{((a,a),a),((a,a),b),((a,b),a),((a,b),b),((b,a),a),((b,a),b),((b,b),a),((b,b),b)\}$
$A \times A^2 = \{(a,(a,a)),(b,(a,a)),(a,(a,b)),(b,(a,b)),(a,(b,a)),(b,(b,a)),(a,(b,b)),(b,(b,b))\}$
$A^3 = \{(a,a,a),(a,b,a),(b,a,a),(b,b,a),(a,a,b),(a,b,b),(b,a,b),(b,b,b)\}$
We can see the three sets have different tuples.
Therefore, we can say $A^{m+n} = A^m \times A^n$ is not true.

Q4

We know the composition of two functions f and g is f(g(x))
The lambda notation for the two functions f and g is: $\lambda(f,g)\cdot\lambda x\cdot f(g(x))$

Q5

(a) 0.01
(b) 0.0397
(c) 0.15407173
(d) 0.17151914210917507
(e) 0.37414651706895796
(f) 1.3056006785398018
(g) 1.3306
(h) This is because there are too many digits.
(i) No, because there will be a round off error.

Q6

$\theta(2^n)$
The output of the star grows by 2^n minus one. In big theta notation, we ignore the constant.

Q7

infinite loop
The inner loop variable j remains the same.

Q8

$\theta(\sqrt{n} \cdot logn)$
The outer loop is going square root of n times, and the the inner loop is going log n times.

Q9

$\theta(n)$
The output is a single star on each line, printing n times.

Q10

$\theta(n^2 logn)$
The outer loop is going n^2 times, and the inner loop is going log n times.

Q11

$\theta(nlogn)$
The outer loop is going n times, and the inner loops is going log n times.

Q12

$\theta(nlogn)$
The outer loop is going log n times, and the inner loop is going n times.

Q13

Best case: $\theta(logn)$
This is because there is still the outer loop even if the if-statement didn't pass.

Worst case: $\theta(n)$
It will take n steps to print out the star in the inner loop.

Q14

Best case: $\theta(1)$
When n is equal to zero, the following operations will not be performed.

Worst case: $\theta(logn)$
The steps will take log n times when executing the operations.

Q15

The time complexity is log 10 $\cdot$ n
Since log 10 is a constant number, the time complexity is just n.

Q16

(a)
T(100) = 100^3 ops
100^3/10 sec = 10^5 ops per second

$T(200) = 200^3 ops$
$200^3/10^5 = 80$ seconds

(b)
$10^5 \cdot 10 \cdot 30 = n^3$
$n = 310$
310 element list

(c)
$(10^{27}/3600)/10^5 = 2.7 \cdot 10^{18}$
$2.7 \cdot 10^{18}$ times faster

Q17

(a)
T(64) = 64 $\cdot$ log 64 ops
64 $\cdot$ log 64/192 = 2 ops per second

$T(128) = 128 \cdot log 128$
$(128 \cdot log 128)/2 = 448$ seconds

(b)
$2 \cdot 8 \cdot 10 =$ n log n
$n = 32$
32 element list

(c)
(10 log 10/1)/2= 16.6096
16.6096 times faster

Q18

(a)
T(32) = 2 ·32· log 32 ops
$(2 \cdot 32 \cdot log 32)/160 = 2$ ops per second

$T(64) = 2 \cdot 64 \cdot log 64$
$(2 \cdot 64 \cdot log 64)/2 = 384$ seconds

(b)
$2 \cdot 4 \cdot 16 = 2 \cdot n \cdot$log n
$n = 16$
16 element list

Q19

n$^2 - 1.1^n = 0$
$n = 95.72$ (by desmos)

Q20

$\lambda n.\frac{2}{n} < \lambda n.log(\sqrt{n}) < \lambda n.2 < \lambda n.2^{log(n)} < \lambda n.2log(n) < \lambda n.\sqrt{n} < \lambda n.nlog(log(n)) < \lambda n.n < \lambda n.nlog(n) < \lambda n.nlog(n^2) < \lambda n.n(log(n))^2 < \lambda n.n^{1.5} < \lambda n.n^2 < \lambda n.n^2 * log(n) < \lambda n.log(n)^n < \lambda n.2^{\frac{n}{2}} < \lambda n.n^3 < \lambda n.n! < \lambda n.n^n < \lambda n.2^{n*n}$