Data Structures and Algorithms I
Spring 2021
Homework #1
Jacob Khalili

100/100

(1) For each of the below GIVEN statements, state whether the following OTHER statement is definitely true, definitely false, or possible (i.e., it could be true or false).

(a) GIVEN: $T_1(N) = O(N)$
OTHER: $T_1(N) = O(N^3)$

Definitely true

(b) GIVEN: $T_1(N) = \Omega(N^4)$
OTHER: $T_1(N) = O(N^6)$

Possible

(c) GIVEN: $T_1(N) = O(N^4 \log N)$
OTHER: $T_1(N) = \Omega(N^5)$

Definitely false

(d) GIVEN: $T_1(N) = O(N^3 \log N)$ and $T_2(N) = O(N^3)$
OTHER: $T_1(N) + T_2(N) = O(N^3)$

Possible

(e) GIVEN: $T_1(N) = O(N^4)$ and $T_2(N) = O(N^5 \log N)$
OTHER: $T_1(N) + T_2(N) = \Theta(N^5)$

Possible

(f) GIVEN: $T_1(N) = o(N)$ and $T_2(N) = o(N^5)$
OTHER: $T_1(N) + T_2(N) = \Theta(N^5)$

Definitely False

(g) GIVEN: $T_1(N) = O(N^3)$ and $T_2(N) = O(N^3 \log N)$
OTHER: $T_1(N) + T_2(N) = O(N^4)$

Definitely true

(h) GIVEN: $T_1(N) = \Omega(N^4)$ and $T_2(N) = \Theta(N^5 \log N)$
OTHER: $T_1(N) + T_2(N) = \Theta(N^5 \log N)$

Possible

Data Structures and Algorithms I
Spring 2021
Homework #1
Jacob Khalili

(i) GIVEN: $T_1(N) = \Theta(N^4)$ and $T_2(N) = \Theta(N^5 \log N)$
OTHER: $T_1(N) + T_2(N) = \Omega(N^{5.1})$

Definitely false

(j) GIVEN: $T_1(N) = \Theta(N^4)$ and $T_2(N) = \Theta(N^5 \log N)$
OTHER: $T_1(N) + T_2(N) = o(N^{5.1})$

Definitely true

50/50

Data Structures and Algorithms I
Spring 2021
Homework #1
Jacob Khalili

(2) Consider the following pseudo-code for a function that takes, as input, a parameter N:

```
Function F(N)
  if F1(N)
    Loop i from 1 to N²
      F2(N)
  else
    F3(N)
    Loop j from 1 to N
      if F4(N)
        F5(N)
      else
        F6(N)
```

For each of the following assumptions, analyze the worst-case running time of Function F using Big-Oh notation, Big-Omega notation, and Big-Theta notation, *if possible*. Express all answers using the tightest possible bounds. *Briefly explain your answers!*

Note: In class, we went over rules related to Big-Oh notation. You will have to infer similar rules for the other notations. Assume that function F1 returns a Boolean value.

(a) Assume that the worst-case running time of F1(N) is O(N), the worst-case running time of F2(N) is O(N), the worst-case running time F3(N) is O(N), the worst-case running time of F4(N) is O(N), the worst-case running time of F5(N) is O(N), and the worst-case running time of F6(N) is O(N).

- O(F(N)) = N^3 – because the first case is the worst possible: $n^2 * n = n^3$
- Ω(F(N)) = N – because F#s are at best constant, and thus with the least about of looping 1*n = n
- Theta ~~may not exist.~~ is unknown

Function:
(A) if $F_1(N)$: → N
Loop i from 1 to N^2
$F_2(N)$ → $N * N^2$) $\underline{N^3}$ $O(N) = \boxed{N^3}$
else
F3(N) → N
Loop j from 1 to N
if F4(N): $N \times \left(N + \frac{N}{2n} \right)$
F5(N)
else:
F6(N)
$N \times (2N)$
$\underline{2N^2 + N}$

Data Structures and Algorithms I
Spring 2021
Homework #1
Jacob Khalili

(b) Assume that the worst-case running time of F1(N) is O(N), the worst-case running time of F2(N) is $O(N^3)$, the worst-case running time F3(N) is $O(N^{11})$, the worst-case running time of F4(N) is $O(N^5)$, the worst-case running time of F5(N) is $O(N^7)$, and the worst-case running time of F6(N) is $O(N^9)$.

- $O(F(N)) = N^{11}$ – because the 2[nd] case is the greatest possible
- $\Omega(F(N)) = N$ – because F#s are at best constant, and thus with the least about of looping 1*n = n
- Can't say anything about Big-Theta

(b)
Function:
if $F_1(N)$: $\longrightarrow N$
Loop i from 1 to N^2
F2(N) $\longrightarrow N^3 \times N^2 = N^5$
else
F3(N) $\longrightarrow N^{11}$
Loop j from 1 to N $\longrightarrow N \times \left(N^5 + \frac{N^7}{N^9} \right)$
if F4(N): $\longrightarrow O(N^5)$
F5(N)
else:
F6(N

$\longrightarrow \boxed{N^{11}}$

(c) Assume that the worst-case running time of F1(N) is Ω(N), the worst-case running time of F2(N) is Ω(N), the worst-case running time F3(N) is Ω(N), the worst-case running time of F4(N) is Ω(N), the worst-case running time of F5(N) is Ω(N), and the worst-case running time of F6(N) is Ω(N).

- $\Omega(N^2)$ – because the lower bond will follow the 2[nd] case
- Can't say anything about Big-O notation, and Big-Theta

(c)
Function:
if $F_1(N)$:) $\longrightarrow N$
Loop i from 1 to N^2
F2(N)) $N \times N^2 = N^3$
else
F3(N) $\longrightarrow N$
Loop j from 1 to N $\longrightarrow$ $N \times \left(N + \frac{N}{N} \right)$
if F4(N):
F5(N)
else:
F6(N

N^3

$(N) \times N = \boxed{N^2} + N$

$\boxed{\Omega = N^2}$

Data Structures and Algorithms I
Spring 2021
Homework #1
Jacob Khalili

(d) Assume that the worst-case running time of F1(N) is Θ(N), the worst-case running time of F2(N) is Θ(N), the worst-case running time F3(N) is Θ(N), the worst-case running time of F4(N) is Θ(N), the worst-case running time of F5(N) is Θ(N), and the worst-case running time of F6(N) is Θ(N).

- $O(N^3)$ (see example a) √
- $\Omega(N^2)$ (see example c) √
- Big-Theta ~~doesn't exist~~ is unknown because they are not equal

(e) Assume that the worst-case running time of F1(N) is Θ(N), the worst-case running time of F2(N) is $\Theta(N^3)$, the worst-case running time F3(N) is $\Theta(N^5)$, the worst-case running time of F4(N) is Θ(N), the worst-case running time of F5(N) is Θ(1), and the worst-case running time of F6(N) is $\Theta(N^2)$.

(e) Function:

```
if F1(N):                     ——o O(N)
    Loop i from 1 to N^2
        F2(N)                 ——o (N^3)(N^2) = [N^5]
else
    F3(N)                     (N^5) +
    Loop j from 1 to N
        if F4(N):             N + [N^5 + N^3]
            F5(N)             1
        else:
            F6(N              (N^2)
```

$O(N^5)$

$\Omega(N^5)$

$\Theta(N^5)$

- All paths lead to a complexity of (N^5) and thus $O(N^5) = \Omega(N^5) = \Theta(N^5)$

√

50/50