1 Asymptotics

Select **all valid** bounds for each implementation of `f0` below. Note that `Math.random` returns a random number between 0 and 1.

(a)
```java
public void f0(int n) {
    if (n < 100) {
        return;
    }
    f0(n - 2);
}
```

☒ $\Omega(1)$	☐ $\Theta(1)$	☐ $O(1)$
☒ $\Omega(\log(n))$	☐ $\Theta(\log(n))$	☐ $O(\log(n))$
☒ $\Omega(n)$	☒ $\Theta(n)$	☒ $O(n)$
☐ $\Omega(n \cdot \log(n))$	☐ $\Theta(n \cdot \log(n))$	☒ $O(n \cdot \log(n))$
☐ $\Omega(n^2)$	☐ $\Theta(n^2)$	☒ $O(n^2)$
☐ $\Omega(n^3)$	☐ $\Theta(n^3)$	☒ $O(n^3)$
☐ $\Omega(3^n)$	☐ $\Theta(3^n)$	☐ The function never
☐ $\Omega(n!)$	☐ $\Theta(n!)$	finishes

(handwritten annotations)

WPL

L
n
n-2
n-4
⋮

100

$\frac{n}{2}$

lower bound
$x \geq 5$
$\geq$ $\geq 4, 3, 2, 1, \ldots$
$x \leq 5$ upper bound
$\leq$ $\leq 6, 7, 8, \ldots$

(b)
```java
public void f0(int n) {
    if (n < 100) {
        return;
    }
    if (Math.random() < 0.5) {
        return;
    }
    f0(n / 2);
}
```

use upper/lower ↙

☒ $\Omega(1)$	☐ $\Theta(1)$	☐ $O(1)$
☐ $\Omega(\log(n))$	☐ $\Theta(\log(n))$	☒ $O(\log(n))$
☐ $\Omega(n)$	☐ $\Theta(n)$	☒ $O(n)$
☐ $\Omega(n \cdot \log(n))$	☐ $\Theta(n \cdot \log(n))$	☒ $O(n \cdot \log(n))$
☐ $\Omega(n^2)$	☐ $\Theta(n^2)$	☒ $O(n^2)$
☐ $\Omega(n^3)$	☐ $\Theta(n^3)$	☒ $O(n^3)$
☐ $\Omega(3^n)$	☐ $\Theta(3^n)$	☐ ~~The function never finishes~~
☐ $\Omega(n!)$	☐ $\Theta(n!)$	

Handwritten notes (right margin):

$2^x = n$

↑ levels = x

$x = \log_2 n$

$= \dfrac{\log n}{\log 2} \cdot 2 \uparrow$

Work constant

$\dfrac{2}{n}$

$\dfrac{n}{2}$

$\dfrac{n}{4}$

$\dfrac{n}{8}$

$\vdots$

→ 100

$\log n$
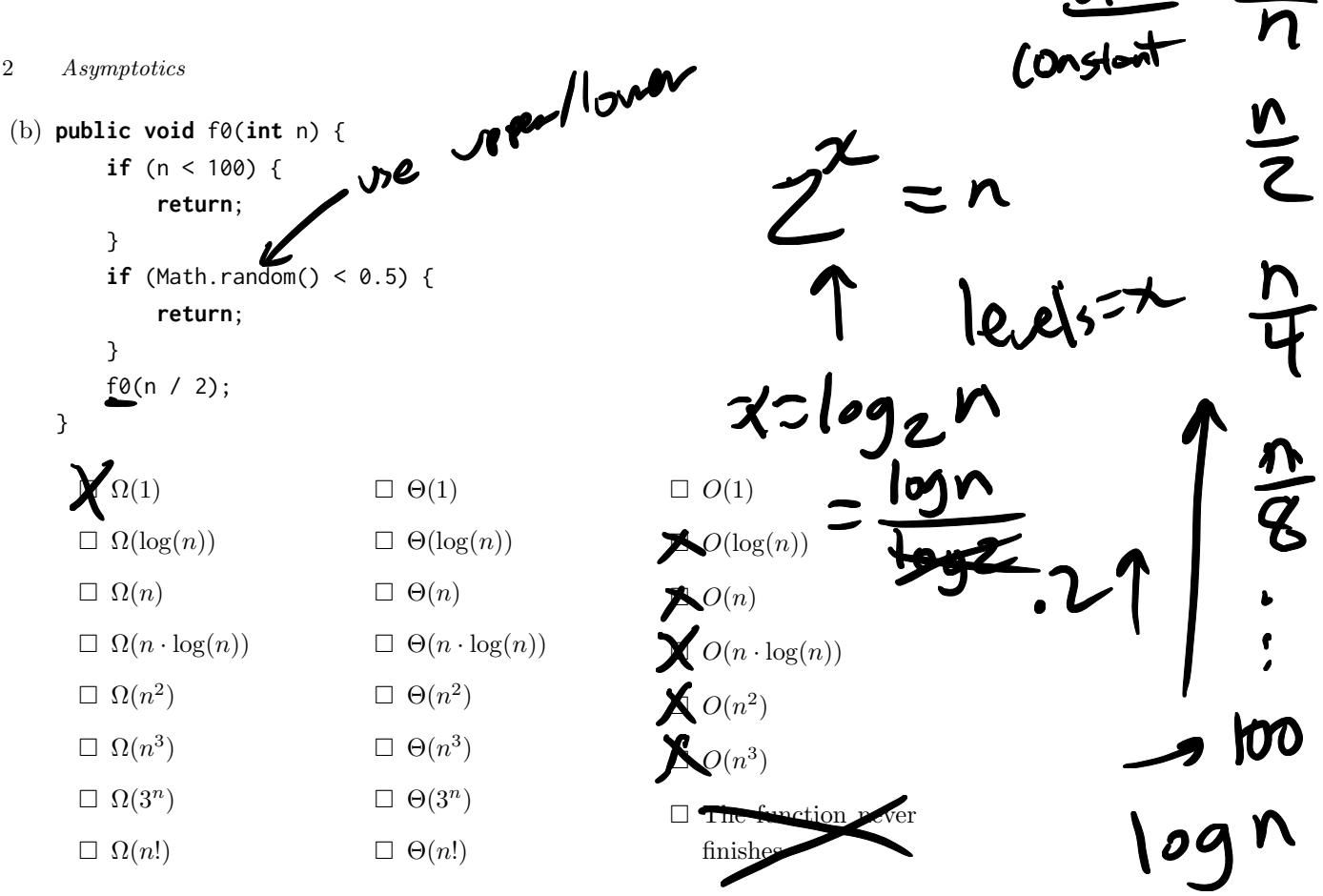
best/worst - cannot assume size of input

2   Best and Worst Case

What is the best and worst case runtimes of the following definitions of `f0` in $\Theta(\cdot)$ notation?
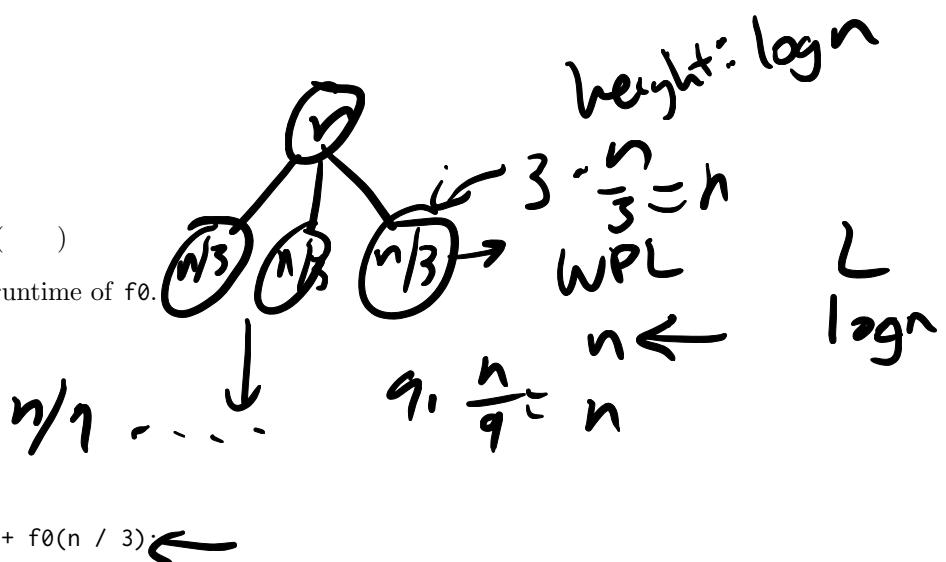
(a)
```java
public int f0(int[] arr) {
    for(int i = 0; i < arr.length; i++) {
        if (arr[i] == 49) {
            return i;
        }
    }
    return -1;
}
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$
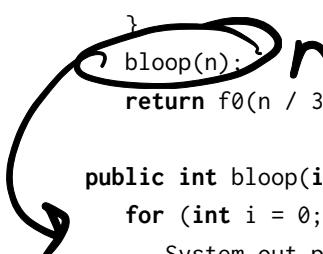
(b) Note that we are trying to find the runtime of `f0`.

```java
public int f0(int n) {
    if (n == 0) {
        return 0;
    }
    bloop(n);
    return f0(n / 3) + f0(n / 3) + f0(n / 3);
}

public int bloop(int n) {
    for (int i = 0; i < n; i += 1) {
        System.out.println("Ah, loops too");
    }
    return n;
}
```

Best Case: $\Theta($ $)$, Worst Case: $\Theta($ $)$