$log(n) \times log(n)$ can be approximated to O(log(n))

Algorithm1 (n)

```
Initialize $i, j, k, count = 0$
for $(i = n/2; i <= n; i = i + 1)$ do
   for $(j = 1; j + n/2 <= n; j = j + 1)$ do
      for $(k = 1; k <= n; k = k \times 2)$ do
         $count \leftarrow count + 1$
      end for
   end for
end for
return count
```

Asymptotic Runtime: ______________________________

Algorithm2 (n)

```
Initialize $i, j, k, count = 0$
for $(i = n/2; i <= n; i = i + 1)$ do
   for $(j = 1; j <= n; j = 2 \times j)$ do
      for $(k = 1; k <= n; k = k \times 2)$ do
         $count \leftarrow count + 1$
      end for
   end for
end for
return count
```

Asymptotic Runtime: ______________________________

Algorithm3 (n)

```
Initialize $i, j, count = 0$
if $(n == 1)$ then
   return
end if
for $(i = 1; i <= n; i = i + 1)$ do
   for $(j = 1; j <= n; j = 2 \times j)$ do
      $count \leftarrow count + 1$
      break
   end for
end for
return count
```

Asymptotic Runtime: ______________________________

PS next page ...

Algorithm4 (n)

```
Initialize $i, j = 1$
while $(i < n)$ do
    $j \leftarrow n$
    while $(j >= 1)$ do
        $j \leftarrow j/2$
    end while
    $i \leftarrow i \times 2$
end while
```

Asymptotic Runtime: ______________________________

Algorithm5 (n)

```
Initialize $i, j, k, count = 0$
for $(i = 1; i <= n; i = i + 1)$ do
    for $(j = 1; j <= n; j = j + 2)$ do
        for $(k = 1; k <= n; k = k + 3)$ do
            $count \leftarrow count + 1$
        end for
    end for
end for
return count
```

Asymptotic Runtime: ______________________________