CSE 332 - Section 3 Worksheet

0. Recurrence Relations

a) Find a recurrence T(n) modeling the worst-case runtime complexity of f(n)

```
1  f(n) {
2    if (n <= 0) {
3      return 1
4    }
5    return 2 * f(n - 1) + 1
6  }
```

b) Find a recurrence T(n) modeling the worst-case runtime complexity of g(n)

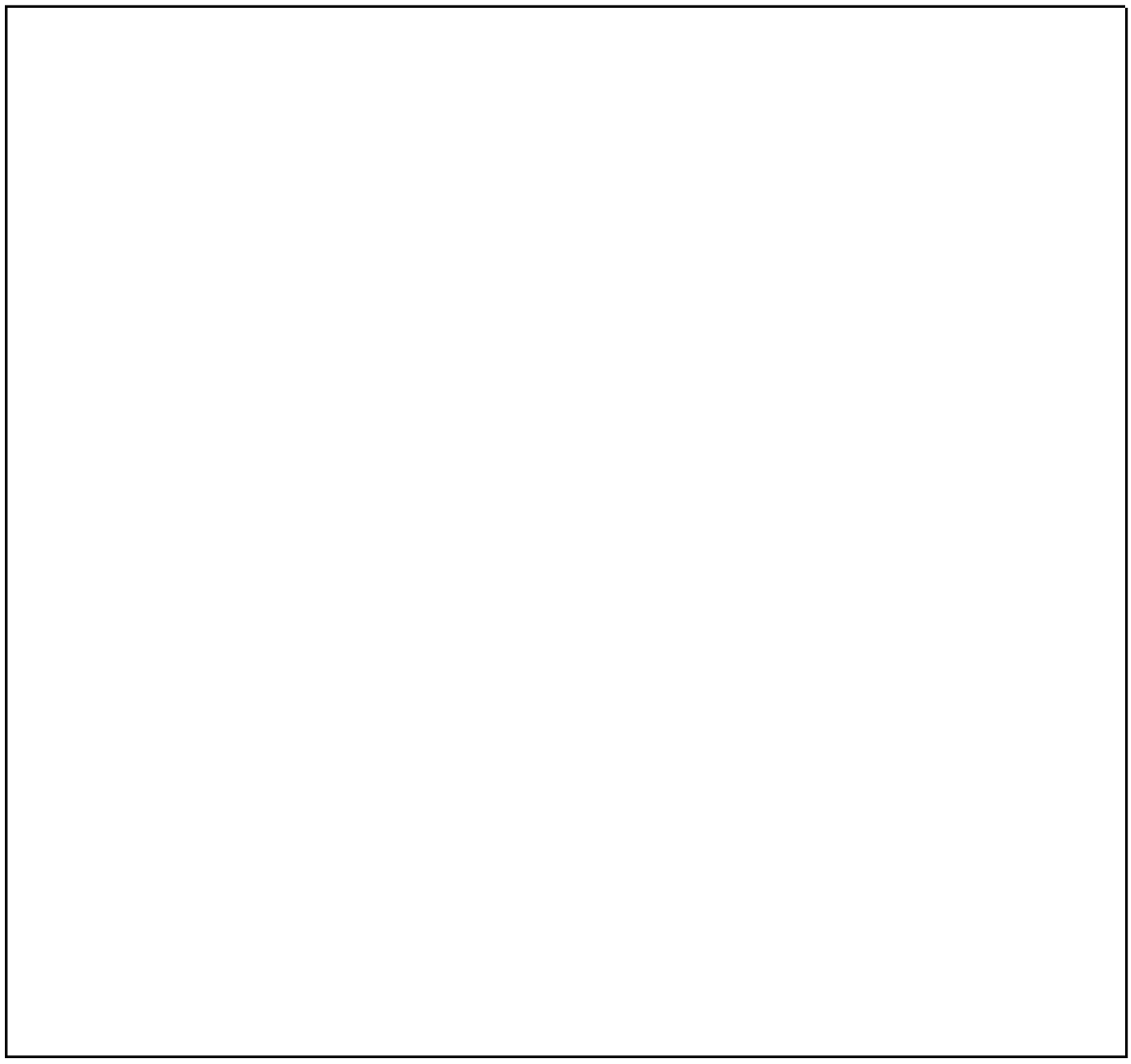
```
1  f(n) {
2    if (n <= 1) {
3      return 1000
4    }
5    if (g(n/3) > 5) {
6      for (int i = 0; i < n; i++) {
7             println("Yay")
8      }
9      return 5 * g(n/3)
10   } else {
11     for (int i = 0; i < n * n; i++) {
12            println("Yay)
13     }
14     return 4 * g(n/3)
15   }
```

CSE 332 - Section 3 Worksheet

1. Tree Method

For each of the following recurrence relations, use the tree method to convert it to closed form:

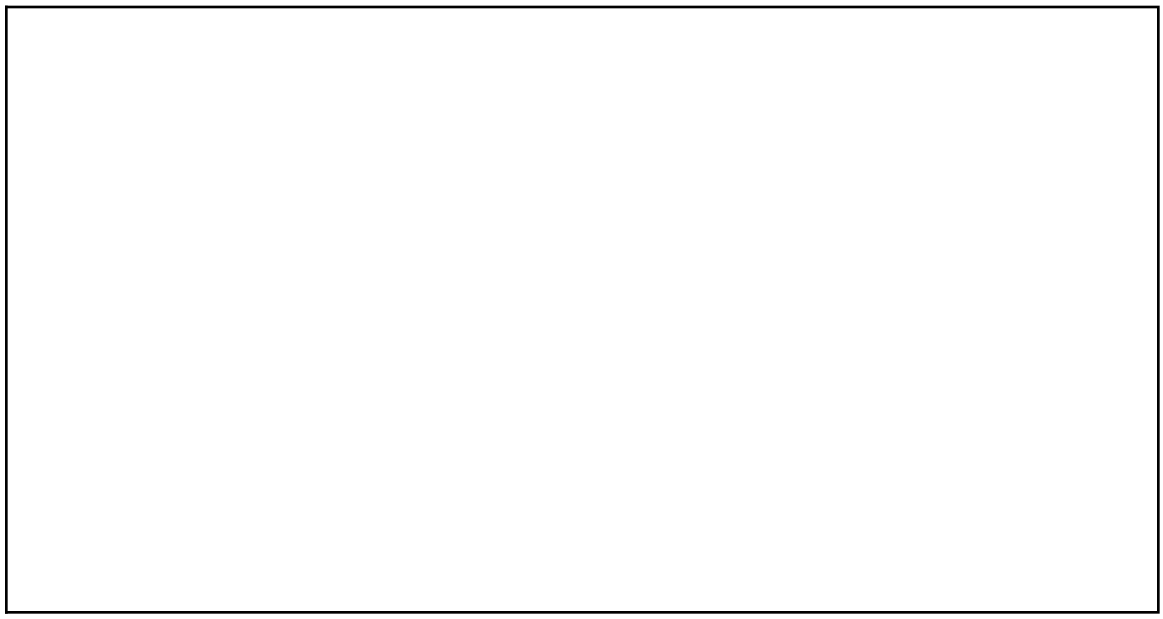
a) $$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 2T\left(\frac{n}{2}\right) + n & otherwise \end{cases}$$



CSE 332 - Section 3 Worksheet

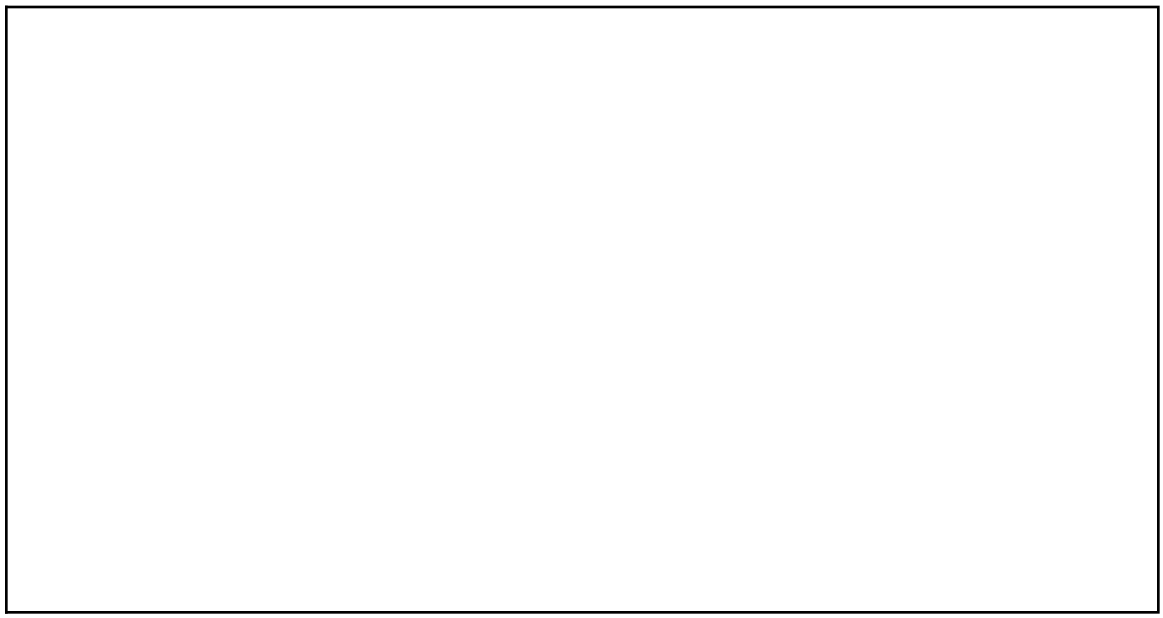
b)

$$T(n) = \begin{cases} 4 & \text{if } n \leq 4 \\ 2T\left(\frac{n}{2}\right) + n & otherwise \end{cases}$$



c)

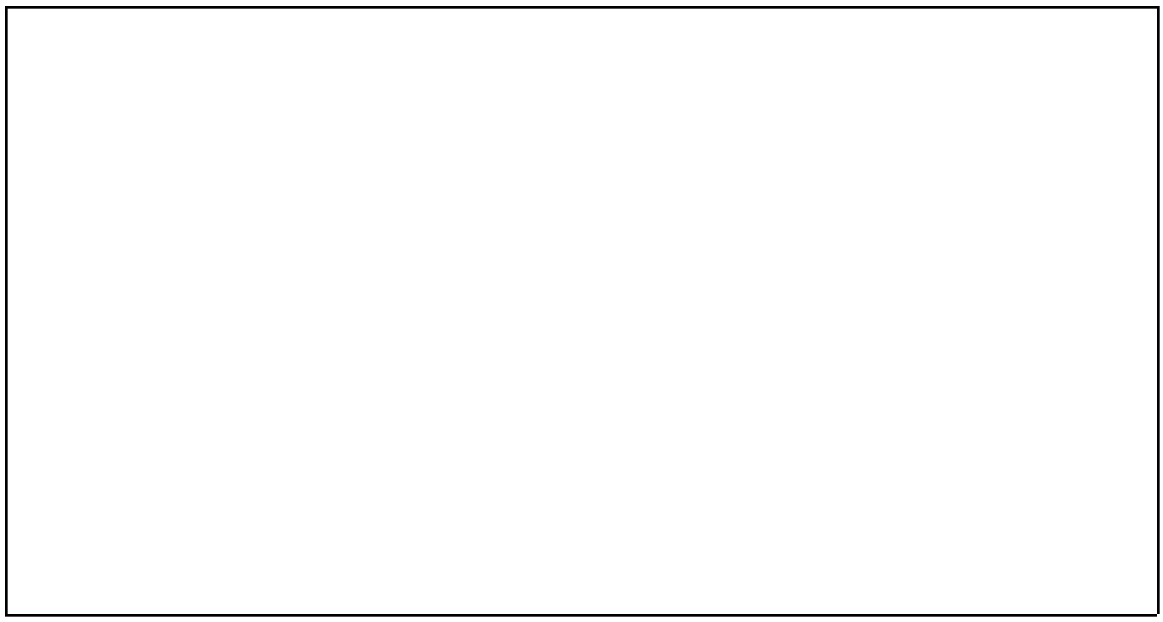
$$T(n) = \begin{cases} 100 & \text{if } n \leq 1 \\ 2T\left(\frac{n}{2}\right) + 100n & otherwise \end{cases}$$



CSE 332 - Section 3 Worksheet

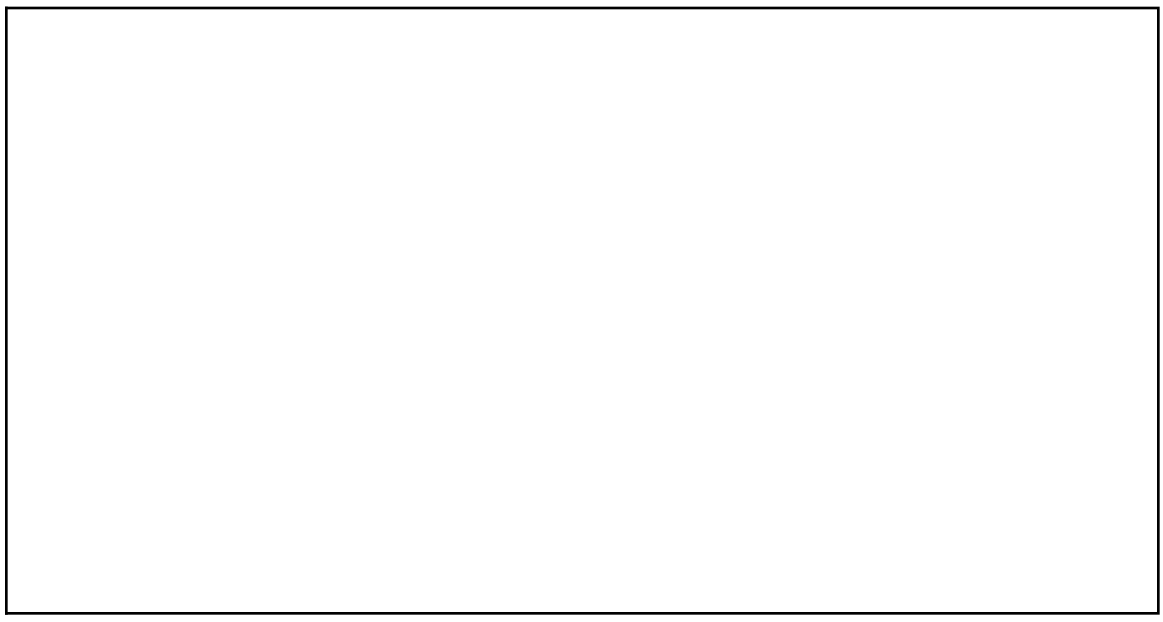
d)

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 3T\left(\frac{n}{2}\right) + n & otherwise \end{cases}$$



e)

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 2T\left(\frac{n}{3}\right) + n & otherwise \end{cases}$$



CSE 332 - Section 3 Worksheet

2. Putting It All Together

Consider the function $f(n)$. Find a recurrence modeling the worst-case runtime of this function and then find a Big-Oh bound for this recurrence.

```
1  f(n) {
2      if (n <= 1) {
3          return 0
4      }
5      int result = f(n/2)
6      for (int i = 0; i < n; i++) {
7          result *= 4
8      }
9      return result + f(n/2)
10 }
```

a) Find a recurrence $T(n)$ modeling the *worst-case runtime complexity* of $f(n)$

b) Use your answer in part (a) to find a closed form for $T(n)$