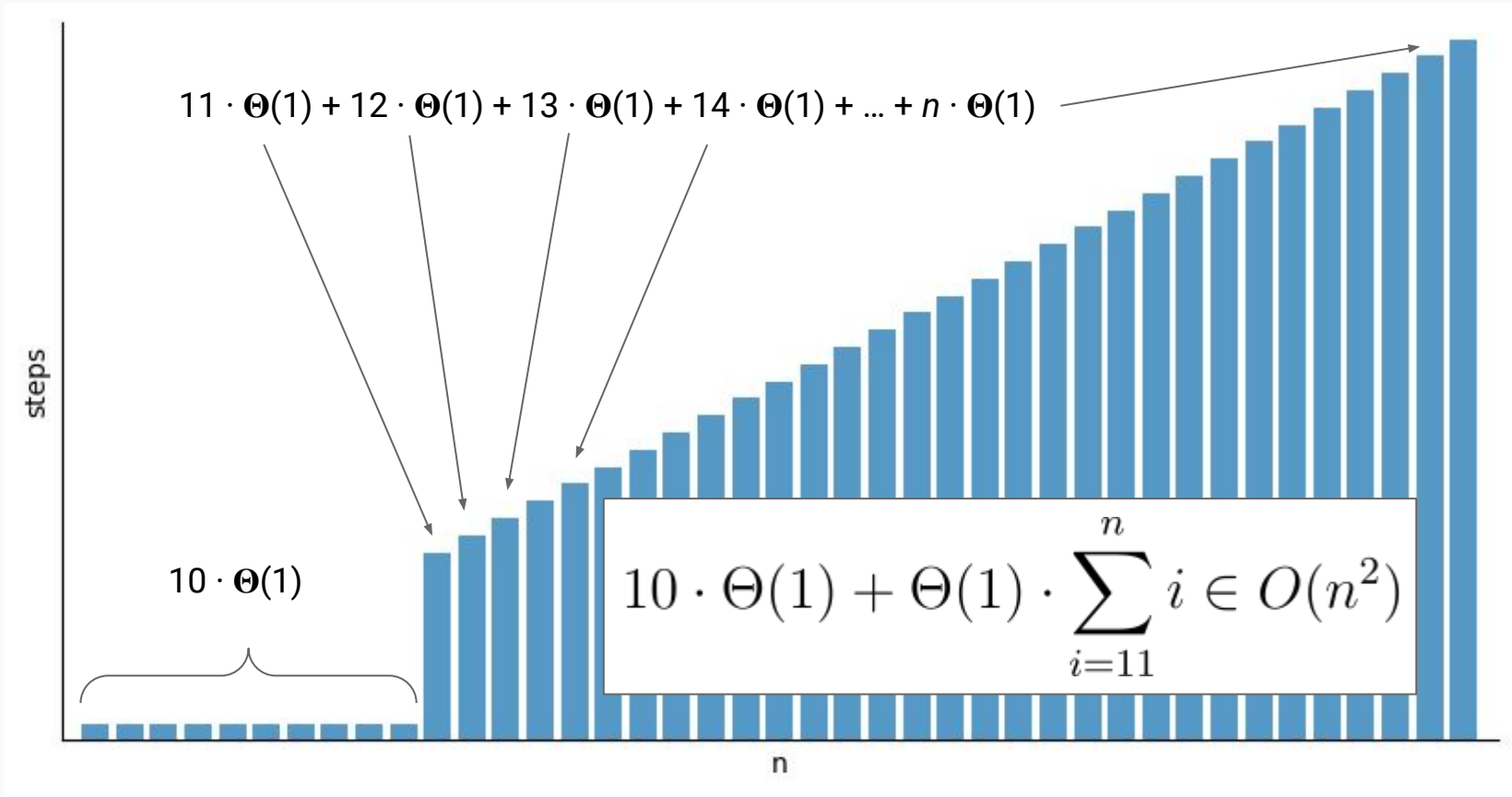# CSE-250 Recitation

Feb 19 - 20 : Amortized Runtime
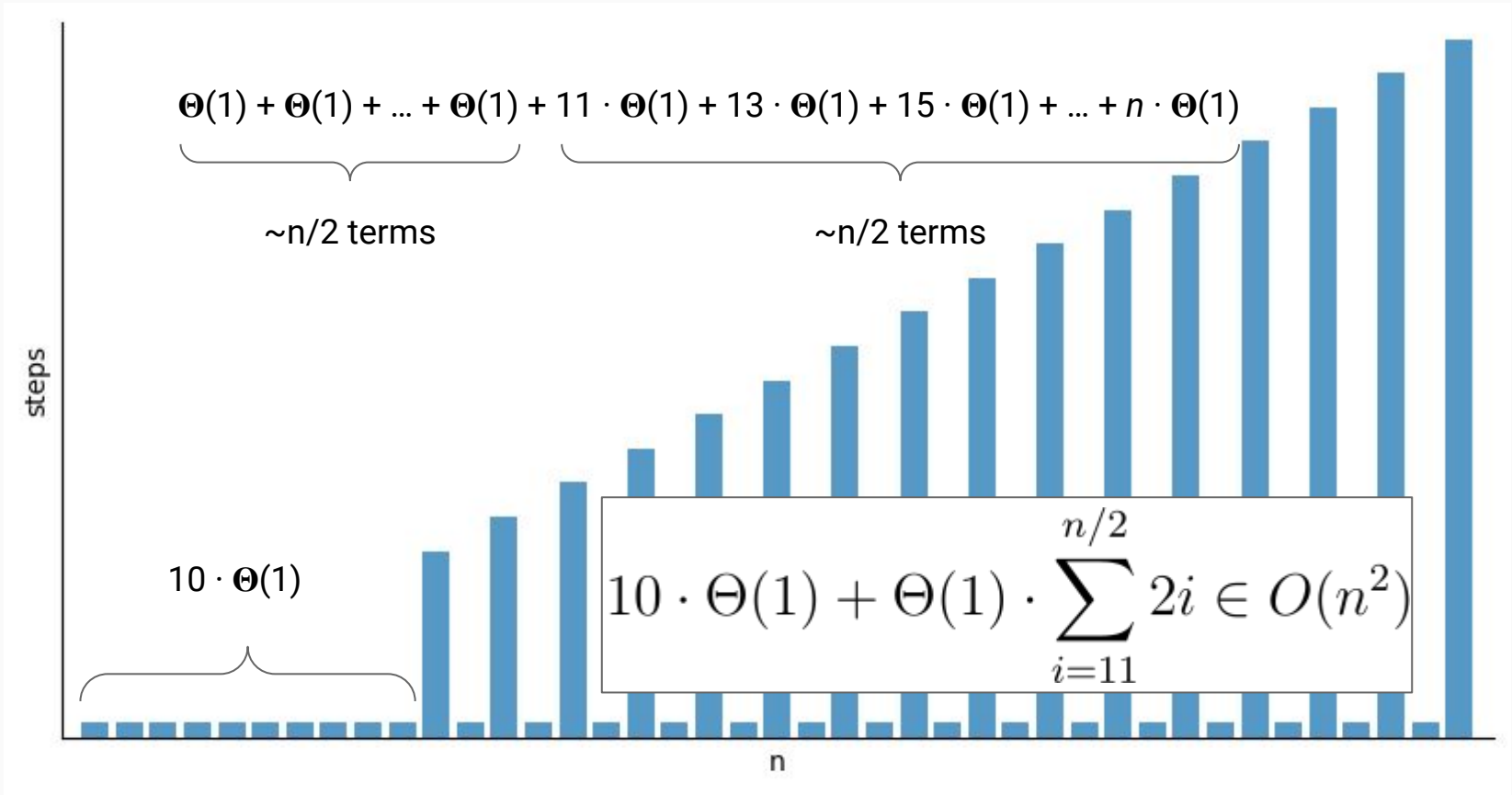
Amortized Runtime

If n calls to a function take $\Theta(f(n))$ steps

Then the **amortized runtime** of that function is $\Theta(f(n)/n)$

$$11 \cdot \Theta(1) + 12 \cdot \Theta(1) + 13 \cdot \Theta(1) + 14 \cdot \Theta(1) + \ldots + n \cdot \Theta(1)$$

$$10 \cdot \Theta(1)$$

$$10 \cdot \Theta(1) + \Theta(1) \cdot \sum_{i=11}^{n} i \in O(n^2)$$

Cost of each call when the initial array size is 10, and newLength = data.length + 1

$$\Theta(1) + \Theta(1) + \ldots + \Theta(1) + 11 \cdot \Theta(1) + 13 \cdot \Theta(1) + 15 \cdot \Theta(1) + \ldots + n \cdot \Theta(1)$$

~n/2 terms

~n/2 terms

$10 \cdot \Theta(1)$

$$10 \cdot \Theta(1) + \Theta(1) \cdot \sum_{i=11}^{n/2} 2i \in O(n^2)$$

steps

n

Cost of each call when the initial array size is 10, and newLength = data.length + 2

$\Theta(1) + \Theta(1) + \ldots + \Theta(1) + 11 \cdot \Theta(1) + 21 \cdot \Theta(1) + 31 \cdot \Theta(1) + \ldots + n \cdot \Theta(1)$

~9/10 · n terms

~1/10 · n terms

10 · $\Theta(1)$

$$10 \cdot \Theta(1) + \Theta(1) \cdot \sum_{i=11}^{n/10} 10i \in O(n^2)$$

steps

n
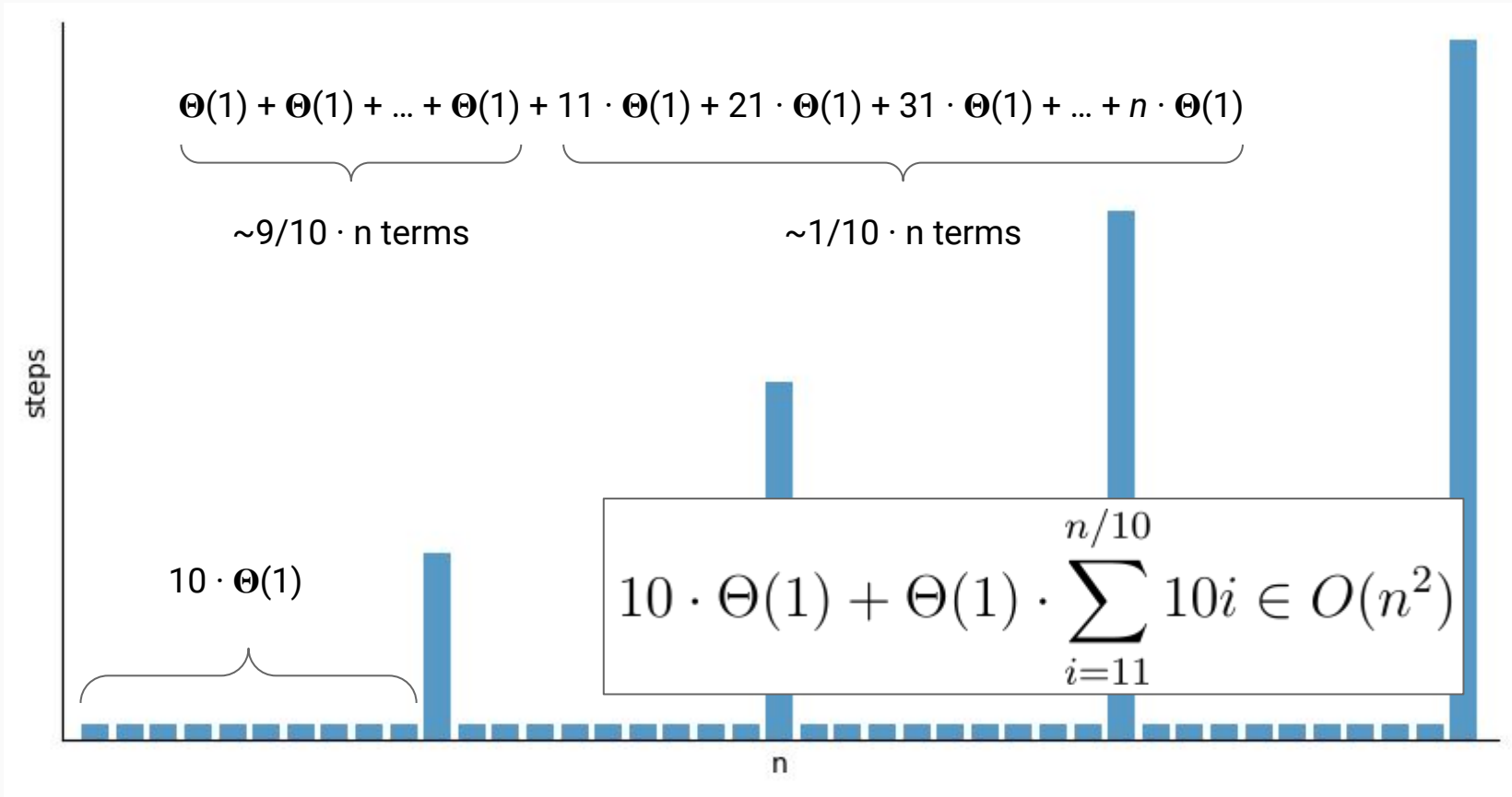
Cost of each call when the initial array size is 10, and newLength = data.length + 10

How can we sum up the total number of steps?
Let I represent the initial size

$2^3I + (2^3I - 1) \cdot \Theta(1)$

$2^2I + (2^2I - 1) \cdot \Theta(1)$

$2^1I + (2^1I - 1) \cdot \Theta(1)$

$2^0I + (2^0I - 1) \cdot \Theta(1)$

$$\sum_{i=0}^{\log(n)} 2^i = O(n)$$

steps

n
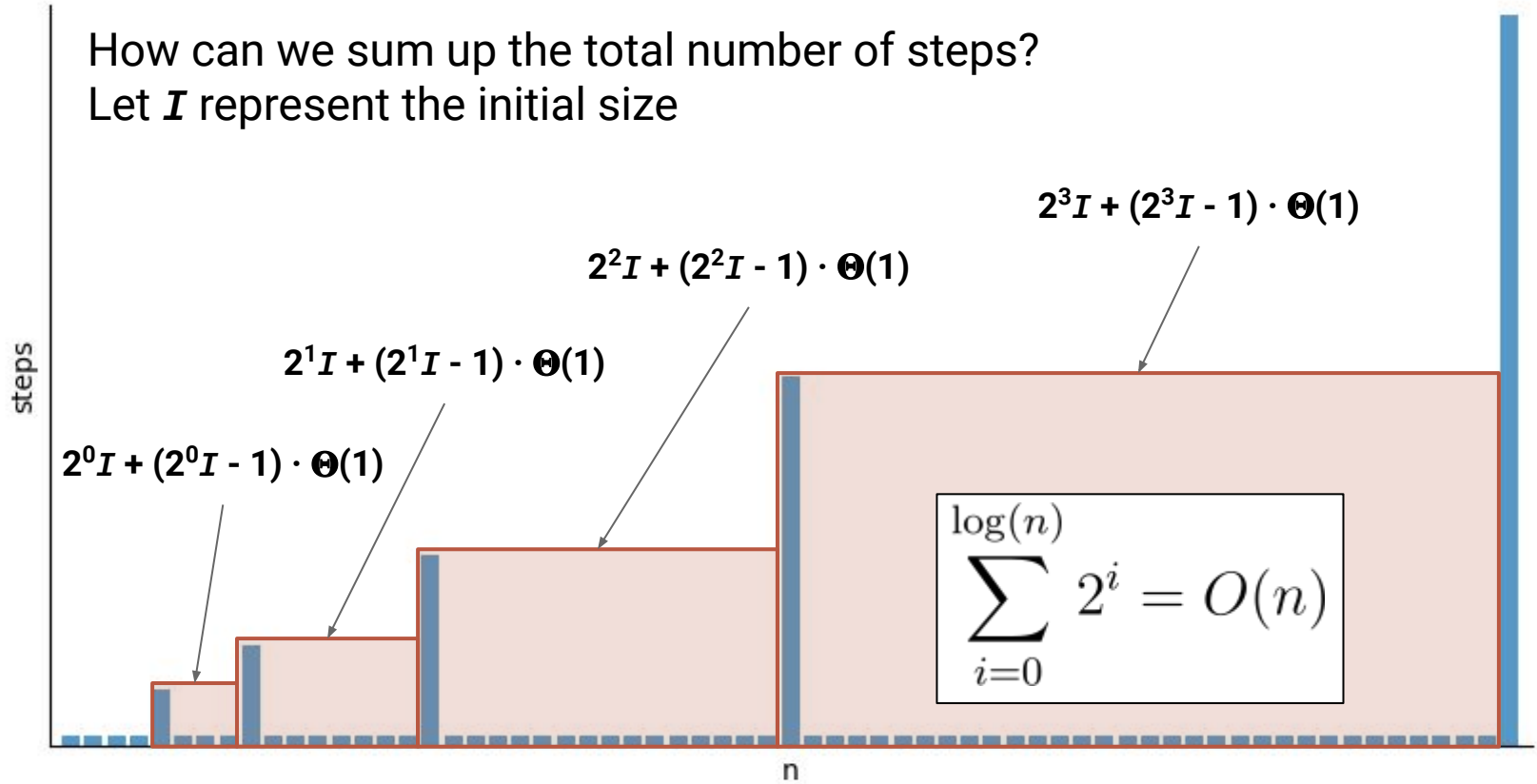
Cost of each call when the initial array size is 4, and newLength = data.length x 2

```java
1 public class Team {
2   private List<Player> players;
3
4   public void addPlayer(Player p) { /* ... */ }
5   public void importRoster(File f) { /* ... */ }
6   /* ... */
7 }
```

```java
1 public void addPlayer(Player p) {
2   System.out.println("Welcome to the team " + p.name());
3   players.add(p);
4 }
```

What are the unqualified and amortized runtime bounds of the **addPlayer** method when **List** is a **LinkedList**? an **ArrayList**?

```
1 public void importRoster(File f) {
2   BufferedReader br = new BufferedReader(new FileReader(f));
3   String line;
4   while (br.ready()) {
5     String line = br.readLine();
6     Player p = new Player(line);
7     players.add(p);
8   }
9 }
```

What are the unqualified and amortized runtime bounds of the `importRoster` method when `List` is a `LinkedList`? an `ArrayList`?

(You can assume that opening the file, reading a line, and creating a `Player` are constant-time calls)