CSE 250 Recitation

April 16 - 17: Induction, Expected Runtime

```
stacks = new Stack()[3]
move(n, from, to):
    if n == 1:
        stacks[to].push(stacks[from].pop())
    else:
        move(n-1, from, other(from,to))
        stacks[to].push(stacks[from].pop())
        move(n-1, other(from,to), to)
```

$$T_{\text{move}}(n) = \begin{cases} \Theta(1) & \text{if } \; n = 1 \\ \Theta(1) + 2 \cdot T_{\text{move}}(n-1) & \text{otherwise} \end{cases}$$

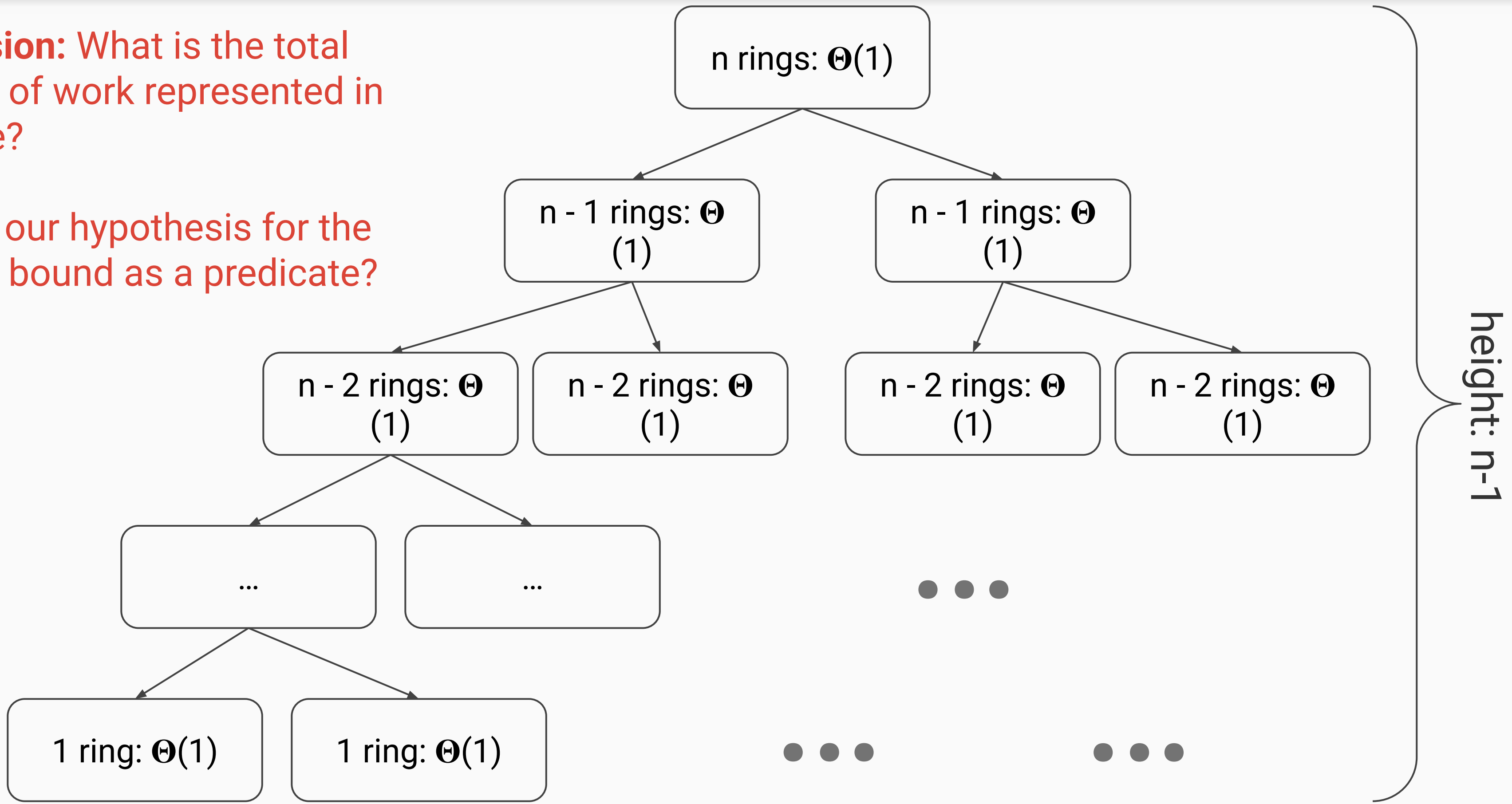
Exercise: Draw the recursion tree for **move**

Label the height in terms of ***n*** and each box with its cost

Recursion Tree for towersOfHanoi

Discussion: What is the total amount of work represented in this tree?

What is our hypothesis for the runtime bound as a predicate?



Hypothesis

Total Work:

A perfect binary tree of height **n-1** has $\mathbf{2^n - 1}$ nodes

Each node is doing $\mathbf{\Theta(1)}$ work

Total work is $\mathbf{\Theta(1) \cdot (2^n - 1)}$

Hypothesis:

$$P(n): T(n) \leq c(2^n - 1)$$

Want to show that for some $\mathbf{c > 0}$, **P(n)** is true for **all** $\mathbf{n \geq n_0}$

Base Case

$$T(n) = \begin{cases} c_0 & \text{if } n = 1 \\ c_1 + 2 \cdot T(n-1) & \text{otherwise} \end{cases}$$

Hypothesis:

P(n): $T(n) \leq c\,(2^n - 1)$

Exercise: Prove the Base Case

Base Case

$$T(n) = \begin{cases} c_0 & \text{if } n = 1 \\ c_1 + 2 \cdot T(n-1) & \text{otherwise} \end{cases}$$

Base Case: $n = 1$

Want to show: **P(1): $T(1) \leq c\,(2^1 - 1)$**

$$c_0 \leq c$$

This is true as long as $c \geq c_0$

Exercise: Do the Induction Step

Hypothesis:

P(n): $T(n) \leq c\,(2^n - 1)$

Induction Step

$$T(n) = \begin{cases} c_0 & \text{if } \; n = 1 \\ c_1 + 2 \cdot T(n-1) & \text{otherwise} \end{cases}$$

Hypothesis:

P(n): $T(n) \leq c\,(2^n - 1)$

Assume: P(k) true for some $k \geq 1$

Show: P(k+1) must be true

Induction Step

$$T(n) = \begin{cases} c_0 & \text{if } n = 1 \\ c_1 + 2 \cdot T(n-1) & \text{otherwise} \end{cases}$$

$T(k+1) \leq c\,(2^{k+1} - 1)$

$c_1 + 2T(k) \leq c\,2^{k+1} - c$

$c_1 + 2\,c(2^k - 1) \leq c\,2^{k+1} - c$

$c_1 + c\,2^{k+1} - 2c \leq c\,2^{k+1} - c$

$c_1 \leq c$

Hypothesis:

P(n): $T(n) \leq c\,(2^n - 1)$

Assume: P(k) true for some $k \geq 1$

Show: P(k+1) must be true

Conclusion

Therefore:

- **P(1)** is true as long as $\mathbf{c \geq c_0}$
- **P(k) ⇒ P(k+1)** is true as long as $\mathbf{c \geq c_1}$

Therefore: If we pick $\mathbf{c = c_1 + c_0}$ (for example), then **P(n)** is true for all $\mathbf{n \geq 1}$

Therefore T(N) $\in O(2^n - 1) = O(2^n)$

Expected Value

A **random variable** represents a quantity that is dependent on random occurrence

Example: Let **X** be the value rolled on a six-sided die

X is a random variable

- It's value can be 1,2,3,4,5 or 6
- It’s value depends on an event (die landing on the “6” face)
- The event depends on a random experiment (rolling the die)

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Example: Let **X** be the value rolled on a six-sided die

Possible values of **X**: 1,2,3,4,5,6

Probability of each outcome: $\frac{1}{6}$

E[X] $= \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = 3.5$

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Generally:

$$E[X] = \sum_i P_i \cdot X_i$$

Probability of the i^{th} outcome

value of the i^{th} outcome

Exercise

Note: A deck of cards contains 52 cards. 4 aces, 4 of each number 2-10, and 12 face cards (4 jacks, 4 queens, 4 kings).

In Blackjack, number cards are worth their value (ie 2 is worth 2, 3 is worth 3, etc), face cards are worth 10, and aces (for simplicity) are worth 1.

If you draw a single card from a shuffled deck of cards, what is the expected value of that card?

$$E[X] = \sum_i P_i \cdot X_i$$

Exercise

Note: $4/52 = 1/13$

If **X** is the value of the drawn card, then

$$E[X] = 1/13 \cdot (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10) + 3/13 \cdot 10 = 6.538$$

Expected Runtime Example #1

```
def mystery(data):
  if randInt() % 100 == 0:
    sum = 0
    for d in data:
      sum += d
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the growth function, T(n), representing the runtime of this function.

What are the unqualified bounds?

Expected Runtime Example #1

```
def mystery(data):
  if randInt() % 100 == 0:
    sum = 0
    for d in data:
      sum += d
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the growth function, T(n), representing the runtime of this function.

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

What are the unqualified bounds?
O(n), Ω(1)

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

The first outcome happens 1/100 times

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

The second outcome happens 99/100 times

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What are the bounds of E[T(n)]?

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What are the bounds of E[T(n)]? **O(n)**

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #2

```
def mystery(data):
  if randInt()%data.size()==0:
    sum = 0
    for d in data:
      sum += d
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the runtime, T(n), and the expected runtime, E[T(n)] for this function.

What are the bounds on these growth functions?

Expected Runtime Example #2

```
def mystery(data):
  if randInt()%data.size()==0:
    sum = 0
    for d in data:
      sum += d
  else:
    sum = data[0] * data.size()
  return sum
```

$$T(n) = \begin{cases} n & \text{if X \% n == 0} \\ 1 & \text{otherwise} \end{cases} \in O(n), \Omega(1)$$

$$E[T(n)] = \frac{1}{n} \cdot n + \frac{n-1}{n} \cdot 1 \in O(1)$$