CSE 250 Recitation

December 1 - 2: Final Review

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
```

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

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

CSE 250 Matrix of ~~Doom~~ Fun

Roll a d6 twice:
https://g.co/kgs/1WcoMvv

The first roll tells you your Data Structure (row)

The second roll tells you your ADT (column)
(if you roll a 6 you get to choose the ADT)

Come up with an implementation for ADT using the Data Structure. Determine the runtime of each key method.

	List	**Stack / Queue**	**Priority Queue**	**Set/Bag**	**Map**
`ArrayList`	Lecture	Lecture		Lecture	
`LinkedList`	Lecture	Lecture	Lecture	Lecture	
`Heap`			Lecture		
`General BST`				Lecture	
`Balanced BST`				Lecture	
`Hash Table`				Lecture	Lecture

CSE 250 Matrix of ~~Doom~~ Fun

Roll a d6 twice:
https://g.co/kgs/1WcoMvv

The first roll tells you your Data Structure (row)

The second roll tells you your ADT (column)
(if you roll a 6 you get to choose the ADT)

Come up with an implementation for ADT using the Data Structure. Determine the runtime of each key method.

	List	Stack / Queue	Priority Queue	Set/Bag	Map
`ArraryList`	Lecture	Lecture		Lecture	
`LinkedList`	Lecture	Lecture	Lecture	Lecture	**TODAY!**
`Heap`			Lecture		
`General BST`				Lecture	
`Balanced BST`				Lecture	
`Hash Table`				Lecture	Lecture

Implementing a `Map` with a `LinkedList`

Discussion: What type of data should the `LinkedList` hold?

```
LinkedList <          > data;

V put(K key, V value):


V get(K key):


V remove(K key):
```

Implementing a `Map` with a `LinkedList`

Discussion: What type of data should the `LinkedList` hold?

Exercise: Implement the remaining `Map` methods

```
LinkedList <Pair<K,V>> data;

V put(K key, V value):


V get(K key):


V remove(K key):
```

Implementing a `Map` with a `LinkedList`

Discussion: What type of data should the `LinkedList` hold?

Exercise: Implement the remaining `Map` methods

```
LinkedList <Pair<K,V>> data;

V put(K key, V value):
    old = remove(key)
    data.add(new Pair(key,value))
    return old

V get(K key):
    for pair in data:
        if pair.key == key:
            return pair.value
    return null

V remove(K key):
    for pair in data:
        if pair.key == key:
            data.remove(pair)
            return pair.value
    return null
```

Blooket Review

https://dashboard.blooket.com/set/6627c6c3a688259d39444174