

Tree Recursion

Announcements

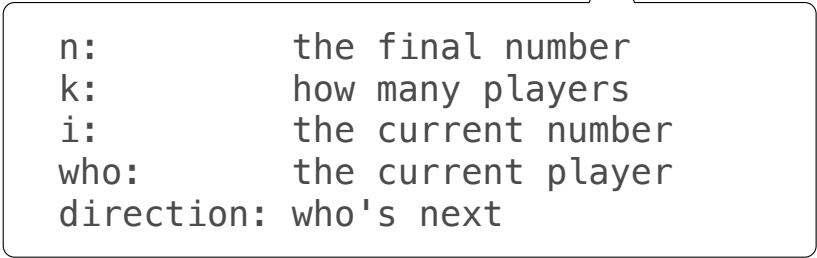
Discussion Review: Sevens

The Game of Sevens

Players in a circle count up from 1 in the clockwise direction. If a number is divisible by 7 or contains a 7 (or both), switch directions. If someone says a number when it's not their turn or someone misses the beat on their turn, the game ends.

Implement `sevens(n, k)` which returns the position of who says `n` among `k` players.

1. Pick an example input and corresponding output.
2. Describe a process (in English) that computes the output from the input using simple steps.
3. Figure out what **additional names** you'll need to carry out this process.
4. Implement the process in code using those additional names.



```
n:          the final number
k:          how many players
i:          the current number
who:        the current player
direction:  who's next
```

(Demo)

Mutual Recursion

Mutually Recursive Functions

Two functions `f` and `g` are mutually recursive if `f` calls `g` and `g` calls `f`.

```
def unique_prime_factors(n):  
    """Return the number of unique prime factors of n.  
  
    >>> unique_prime_factors(51) # 3 * 17  
    2  
    >>> unique_prime_factors(9)  # 3 * 3  
    1  
    >>> unique_prime_factors(576) # 2 * 2 * 2 * 2 * 2 * 2 * 3 * 3  
    2  
    """
```

(Demo)

Tree Recursion

Spring 2023 Midterm 2 Question 5

Definition. When parking vehicles in a row, a motorcycle takes up 1 parking spot and a car takes up 2 adjacent parking spots. A string of length n can represent n adjacent parking spots using % for a motorcycle, <> for a car, and . for an empty spot.

For example: `'.%%.<><>'` (Thanks to the Berkeley Math Circle for introducing this question.)

Implement `count_park`, which returns the number of ways that vehicles can be parked in n adjacent parking spots for positive integer n . Some or all spots can be empty.

```
def count_park(n):
    """Count the ways to park cars and motorcycles in n adjacent spots.
    >>> count_park(1) # '.' or '%'
    2
    >>> count_park(2) # '.. ', '.%', '%.', '%%', or '<>'
    5
    >>> count_park(4) # some examples: '<><>', '.%%.', '%<>%', '%.<>'
    29
    """
    if n < 0:
        return ____
    elif n == 0:
        return ____
    else:
        return ____
```


Preview: Memoization

A *memoized* function stores the return value for every argument it receives. When called a second time with the same argument, it returns the stored value rather than recomputing it.

LRU: least recently used

Cache: stored arguments and their return values

```
from functools import lru_cache
memoize = lru_cache(None)
```

```
@memoize
def f(n):
    ...
```

Store all return values instead of just a limited number

Soon we will be able to implement @memoize ourselves.

(Demo)

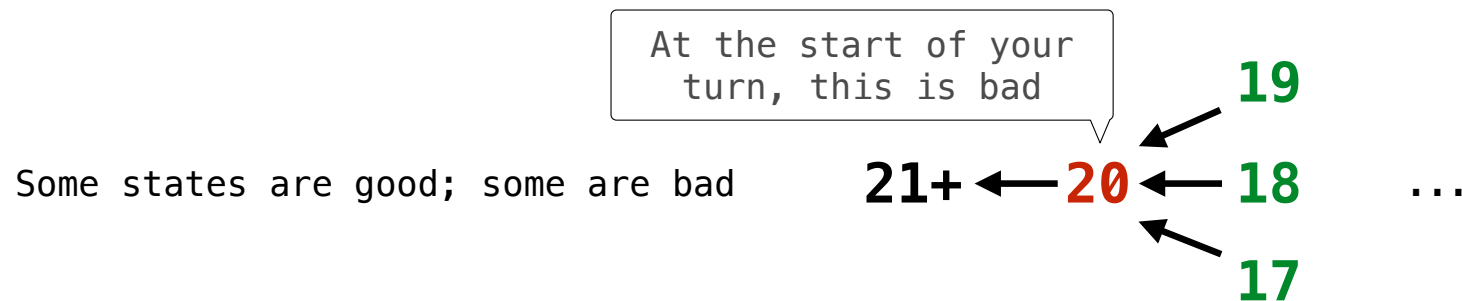
Twenty-One Rules

Two players alternate turns, on which they can add 1, 2, or 3 to the current total

The total starts at 0

The game end whenever the total is 21 or more

The last player to add to the total loses



(Demo)