**Problem #1688: Count of matches in Tournament**

<https://leetcode.com/problems/count-of-matches-in-tournament/description/>

**My Solution:**

**Solution 1:**

In each round of the tournament, the teams become half until there is a winner.

1. Initialize numMatches to 0, teams to n and rounds to 0.
2. While teams is greater than 1, numMatches will be teams divided by 2 (integer division).

If number of teams is odd, teams remaining will be teams divided by 2 (integer division) plus 1.

If number of teams is even, then teams remaining will be teams divided by 2.

Increment rounds by 1.

1. Return number of matches.

class Solution:

def numberOfMatches(self, n: int) -> int:

numMatches = 0

teams = n

rounds = 0

while teams > 1:

numMatches += teams//2

if teams % 2 == 1:

teams = (teams//2) + 1

else:

teams = teams//2

rounds += 1

#print("After rounds = ", rounds, "teams = ", teams, "numMatches = ", numMatches)

return numMatches

**Solution 2:  
  
Mathematical Solution:**

When 2 teams play, only 1 team gets eliminated. To have a winner among n teams, (n – 1) teams should be eliminated. So for (n – 1) teams to be eliminated, there should be (n – 1) matches.

class Solution:

def numberOfMatches(self, n: int) -> int:

return n - 1

**See Degroot Probability & Statistics, Chapter 1, Tennis Tournament Problem.**