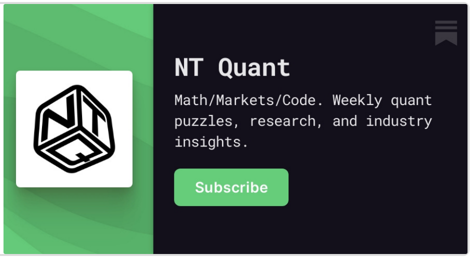
Quant Puzzles 9: Pirate Problems

NT Quant | Substack

Math/Markets/Code. Weekly quant puzzles, research, and industry insights. Click to read NT Quant, a Substack publication with thousands of readers.

http://ntquant.substack.com


NT Quant
Math/Markets/Code. Weekly quant puzzles, research, and industry insights.
Subscribe

Problem 1: Screwy Pirates [Easy]

5 pirates have looted 100 coins. They must vote on how to divide them amongst each other. The voting process is the following:

- Senior most pirate proposes a split
- If the senior-most pirate's plan gets at least 50% approval (he can vote for himself), the plan is passed. Otherwise he is executed and the process repeats with the next senior-most pirate.
- Assume the following:
  - All pirates are perfectly rational and place the highest priority on staying alive
  - All pirates seek to maximize earnings
  - The pirates are blood-thirsty and will reject a plan if it doesn't put their life in danger or cause them to earn less money

Solution:

This problem is recursive in nature.

A common technique for problems is to reduce them to a base case and then gradually build up. Let n be the number of pirates.

$n = 1$: With only 1 pirate, he gets 100 coins

$n = 2$: The senior most pirate votes for himself and gets 100 coins

$n = 3$: The senior most pirate gives himself 99 coins and gives 1 coin to pirate #1, who gets nothing if he doesn't vote to approve the plan.

$n = 4$: The senior most pirate gives himself 99 coins and gives 1 coin to pirate #2, who gets nothing if he doesn’t vote in favor of the plan.

$n = 5$: The senior most pirate gives himself 98 coins, and 1 coin to pirate #3, and pirate #1 each of whom get nothing if they do not vote to approve the plan.

We can notice a pattern, at each iteration the pirate distributes 1 coin to each of the pirates who get nothing in the previous allocation (until he secures floor(n/2) votes)

Problem 2: Generalized Screwy Pirates [Medium]

Write a program that outputs exactly how much gold each pirate gets for n pirates and k coins such that $n \leq k$.

Solution:

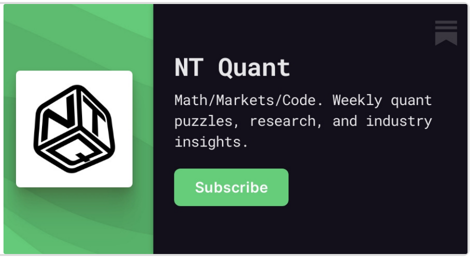
We can formulate this puzzle as a dynamic programming problem. There is a finite number of coins k. For some number of pirates **n**, the senior-most pirate gives one coin to any pirate who does not receive any coins from the **n-1** case (which buys him a vote) until he secures the necessary number of votes. He will then give the remainder of the coins to himself.

```
def screwy_pirates(n,k):
    #nxn grid
    dp = [[0 for i in range(n)] for j in range(n)]
    dp[0][0] = k
    for i in range(1,n):
        #starts with 1 vote since he will vote for himself
        votes = 1
        coins = k
        for j in range(i):
            if dp[i-1][j] == 0 and votes < (n/2):
                dp[i][j] = 1
                coins-=1
                votes+=1
        dp[i][i] = coins
    return dp[n-1]
```

NT Quant | Substack

Math/Markets/Code. Weekly quant puzzles, research, and industry insights. Click to read NT Quant, a Substack publication with thousands of readers.

http://ntquant.substack.com



NT Quant

Math/Markets/Code. Weekly quant puzzles, research, and industry insights.