AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Name: Pearl Nibebadaar Sandoh Kuuridong — Assignment Number: 1
Course: Mathematical Problem Solving — Date: September 6, 2025

Exercise 2

The game we had on Tuesday in class, known as Wythoff's Nim is played according to the following rules:

1. Two players start with any couple of positive integers. That is $(m, n) \quad m, n \in \mathbb{Z}^+$.

2. Each player should subtract any positive integer k from (m, n) when it's their turn to play. If a player chooses to subtract from m then k must be less than or equal to m likewise, if a player chooses to subtract from n then k must be less than or equal to n. A player is allowed to subtract from both m and n at the same time, in which case the same value of k must be subtracted from both sides. In that case $k \leq \min\{m, n\}$

3. The game ends when we reach $(0, 0)$ and the player who subtracted the values to reach $(0, 0)$ wins the game.

From the rules described, we see that since $(0, 0)$ is a winning position, it is always possible to reason backwards to generate some other positions (m, n) for which $(0, 0)$ will always be the outcome, provided the player who played (m, n) plays to win the game on their subsequent moves.

Thus per the rules of the game $(m, 0), (m, m), (0, m)$, for some positive integer m, will always lead to $(0, 0)$ since the next player can always subtract m from either the left or the right side of the couple and generate $(0, 0)$ or subtract m from both sides to still result in $(0, 0)$. Since m must be a positive integer, we consider the smallest possible positive integer, 1, and find the moves that can lead to $(1, 0), (0, 1), (1, 1)$.

Then any couple of the form $(t, 1) \quad t \in \mathbb{N}, t \neq 1$ can always lead to any of $(1, 0), (0, 1), (1, 1)$. To ensure that $(t, 1) \quad t \in \mathbb{N}, t \neq 1$ always leads to any of $(1, 0), (0, 1),$ or $(1, 1)$ then per the rules of the game, where we have that the positive integer k that we subtract from one side (m, n) must be less than or equal to the minimum $\min\{m, n\}$, $t = 2$ if we consider the case where we want to subtract from both sides at the same time. So we want to test if $(1, 2)$ is a winning position. Notice that, assuming perfect play, $(1,2)$ will always lead to one of $(1,0),(0,1),(1,1)$. So our next winning position is $(1,2)$

Since we have found another winning position $(1,2)$, then we can use that to find subsequent winning positions. To find numbers that can lead to $(1,2)$, according to the rules of the game, then we must find some (m,n) that can lead to $(1,2)$ assuming optimal play

Case 1: When we subtract k from m only
Then we know that $k \leq m$ and $n = 2$. So $(m-k, 2) = (1,2)$. Thus $m - k = 1 \iff m = k+1$. Since k is a positive integer and (m,m) leads directly to $(0,0)$ (assuming perfect play) then $k \in \{2,3,4,\ldots\}$
Thus we have values of the form $(3,2),(4,2),(5,2),\ldots$ that can lead to $(1,2)$ assuming optimal play.

Case 2: When we subtract k from n only
Then we know that $k \leq n$ and $m = 1$. So $(1, n-k) = (1,2)$. Thus $n - k = 2 \iff n = k+2$. Since k is a positive integer then $k \in \{1,2,3,\ldots\}$
Thus we have values of the form $(1,3),(1,4),(1,5),\ldots$ that can lead to $(1,2)$ assuming optimal play.

Case 3: When we subtract k from both sides
Then we know that $k \leq \min\{m,n\}$ and $(m-k, n-k) = (1,2)$ thus $(m,n) = (m, m+1), \quad m \in \{2,3,4,\ldots\}$

Thus we have that, assuming perfect play, values of the form $(1,s),(t,2),(r,r+1) \quad r \in \{2,3,4,\ldots\}, s \in \{3,4,5,\ldots\}, t \in \{2,3,4,\ldots\}$ will always lead to the winning position $(1,2)$.

Now to since we know that $(1,2)$ is a winning position then we find (m,n) that can always lead to $(1,2)$, assuming optimal play, we take all positions (m,n) such that either

1. $m-k$ equals the first coordinate of a previous winning position and n equals the second coordinate, or

2. $n-k$ equals the second coordinate and m equals the first coordinate, or

3. $m-k$ and $n-k$ equal the coordinates of a previous winning position,

for some positive integer k. That is
Subtract from m only: $(m-k, n) = (1,2)$ gives $m-k = 1$ and $n = 2$, which leads to $m = k+1 > 1$ and $n = 2$. The minimal choice $k = 2$ yields $(m,n) = (3,2)$, which corresponds to $(2,3)$ after ordering. This cannot be a winning position since it is of the form $(r, r+1)$ which leads directly to $(1,2)$ assuming optimal play.

Subtract from n only: $(m, n-k) = (1,2)$ gives $n-k = 2$ and $m = 1$, leading to $n = k+2$. The minimal choice $k = 1$ yields $(m,n) = (1,3)$. This is also not a winning position since it is of the form $(1,s), s \in \{3,4,5,\ldots\}$ which leads directly to $(1,2)$ assuming optimal play.

Subtract from both: $(m-k, n-k) = (1,2)$ requires $m-k = 1$ and $n-k = 2$, which implies $n - m = 1$ and $m > 1$. Choosing $m = 3$ gives $n = 5$, producing $(3,5)$. Which becomes our next winning position.

Following the same procedure we can find the first n winning positions. In particular, the first 10 winning positions are

$$(0,0),(1,2),(3,5),(4,7),(6,10),(8,13),(9,15),(11,18),(12,20),(14,23)$$

Notice that starting from $(1,2)$ the absolute difference between terms (m,n) is $\mid m-n \mid$ and that forms a monotonic increasing sequence of consecutive non negative integers:

$$\begin{aligned}
(0,0) &: |0-0| = 0 \\
(1,2) &: |2-1| = 1 \\
(3,5) &: |5-3| = 2 \\
(4,7) &: |7-4| = 3 \\
(6,10) &: |10-6| = 4 \\
(8,13) &: |13-8| = 5 \\
(9,15) &: |15-9| = 6 \\
(11,18) &: |18-11| = 7 \\
(12,20) &: |20-12| = 8 \\
(14,23) &: |23-14| = 9
\end{aligned} \tag{1}$$

Notice also, that each natural number appears only once on either the left side of the couple or the right side of the couple but not both sides. And the term on the left side of the next couple is the smallest unused natural number that has not yet appeared in the sequence. We call $(0,0)$ the first winning position. From 1 it is easy to conclude that all the terms on the right side of each couple is the sum of the terms on the right and the difference between the two terms. Still from 1 we can also see that the tith winning position is given by $i-1$. So now to find the ith winning position, we can form a strategy like so:

1. If $i=1$, then $(0,0)$ is your winning postion (stopping criteria for game).

2. If $i>1$,
   (a) Find all winning positions up to the $(i-1)$th position.
   (b) Set $m=\min\{j\}$ where j is the sequence of all the non-negative integers that have not been used in any of the previous winning positions.
   (c) Set $n=m+(i-1)$
   (d) (m,n) is your next winning position.

This strategy will get you the first n winning positions for some natural number n

The code implementation for this strategy is

```
def win_pos(n):
  if n==1:
```

```
    return (0,0)
  positions = [(0,0)]

  start_m, start_n = (0, 0)
  used_ints = {start_m}

  for common_diff in range(1, n):
    m = 0
    while m in used_ints:
      m+= 1
    if m not in used_ints:
      y = m + common_diff
      used_ints.add(m)
      used_ints.add(y)
      positions.append((m,y))
  return positions
```

Calling `print(win_pos(10))` returns

```
[(0, 0),
(1, 2),
(3, 5),
(4, 7),
(6, 10),
(8, 13),
(9, 15),
(11, 18),
(12, 20),
(14, 23)]
```

Now since we are able to generate the first n winning positions for some $n \in \mathbb{N}$ then we shall proceed plot the first 100 points on a graph to check the behavior of the sequence of winning points. The code to generate the plot is given below:

```
import matplotlib.pyplot as plt
values = win_pos(100)
x = [values[pos][0] for pos in range(len(values))]
y = [values[pos][1] for pos in range(len(values))]

plt.plot(x, y)
```

Generates the following:

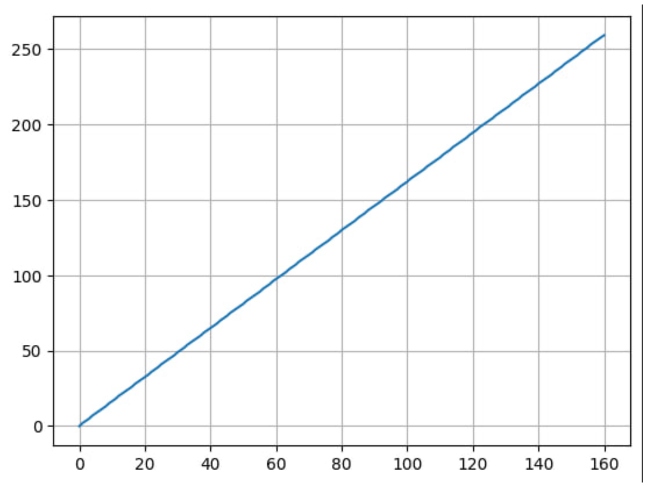

Figure 1: Graph of first 100 winning positions (m, n)

Since the graph of the winning positions is a straight line and lines are uniquely determined by their slopes and intercepts. Then we will find the intercept and slope of the line formed by the first few winning positions.

```
import numpy as np
slope, intercept = np.polyfit(x, y, 1)
print(slope)
```

This returns 1.6180499523215728. Notice that this number has the same first few numbers as $\phi = \frac{1+\sqrt{5}}{2}$ and since we already noticed that the absolute difference in the terms of the winning position form a sequence of consecutive non-negative integers starting from 0, then we can explore how that combines with the $m = 1.6180499523215728 \approx \phi = \frac{1+\sqrt{5}}{2}$. In particular, notice that

$$
\begin{aligned}
&(0 \cdot \phi,\ 0 \cdot \phi^2) = (0, 0)\\
&(1 \cdot \phi,\ 1 \cdot \phi^2) = (1.618033989,\ 2.618033989) \approx (1, 2) = (\lfloor 1.618033989 \rfloor,\ \lfloor 2.618033989 \rfloor)\\
&(2 \cdot \phi,\ 2 \cdot \phi^2) = (3.236067978,\ 5.236067978) \approx (3, 5) = (\lfloor 3.236067978 \rfloor,\ \lfloor 5.236067978 \rfloor)\\
&(3 \cdot \phi,\ 3 \cdot \phi^2) = (4.854101967,\ 7.854101967) \approx (4, 7) = (\lfloor 4.854101967 \rfloor,\ \lfloor 7.854101967 \rfloor)\\
&(4 \cdot \phi,\ 4 \cdot \phi^2) = (6.472135956,\ 10.472135956) \approx (6, 10) = (\lfloor 6.472135956 \rfloor,\ \lfloor 10.472135956 \rfloor)\\
&(5 \cdot \phi,\ 5 \cdot \phi^2) = (8.090169945,\ 13.090169945) \approx (8, 13) = (\lfloor 8.090169945 \rfloor,\ \lfloor 13.090169945 \rfloor)\\
&(6 \cdot \phi,\ 6 \cdot \phi^2) = (9.708203934,\ 15.708203934) \approx (9, 15) = (\lfloor 9.708203934 \rfloor,\ \lfloor 15.708203934 \rfloor)\\
&(7 \cdot \phi,\ 7 \cdot \phi^2) = (11.326237923,\ 18.326237923) \approx (11, 18) = (\lfloor 11.326237923 \rfloor,\ \lfloor 18.326237923 \rfloor)\\
&(8 \cdot \phi,\ 8 \cdot \phi^2) = (12.944271912,\ 20.944271912) \approx (12, 20) = (\lfloor 12.944271912 \rfloor,\ \lfloor 20.944271912 \rfloor)\\
&(9 \cdot \phi,\ 9 \cdot \phi^2) = (14.562305901,\ 23.562305901) \approx (14, 23) = (\lfloor 14.562305901 \rfloor,\ \lfloor 23.562305901 \rfloor)
\end{aligned}
$$

So we see a pattern which is to find the ith winning position, we can make use of the general formula $(\lfloor (i-1)\phi \rfloor, \lfloor (i-1)\phi^2 \rfloor)$. Now to find the general strategy for winning, notice that for any winning position (m, n), (n, m) is also a winning position since the piles are the same, just swapped, and so a player can take away tokens the same way. So since we have that $(\lfloor (i-1)\phi^2 \rfloor, \lfloor (i-1)\phi \rfloor)$ also gives winning positions then this sequence of winning positions must also form a line.

```
import numpy as np
import matplotlib.pyplot as plt
values = win_pos(100)
x = [values[pos][0] for pos in range(len(values)) ]
y = [values[pos][1] for pos in range(len(values)) ]

new_slope, new_intercept = np.polyfit(y, x, 1)
print(new_slope)
plt.plot(y,x)
```

This returns a slope of 0.6180244133494199 and the following plot

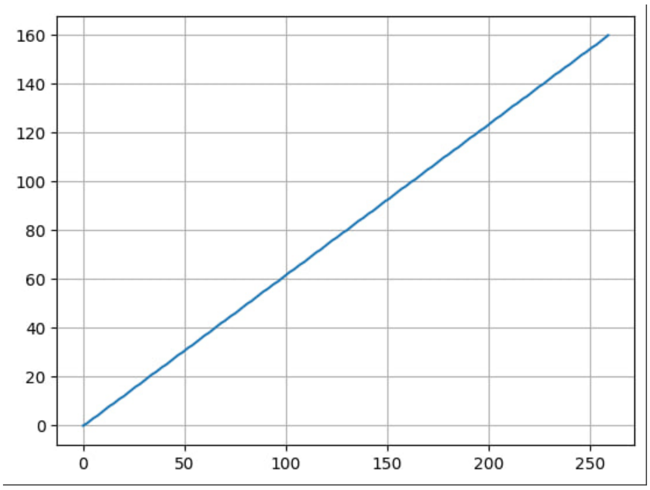

Figure 2: Graph of first 100 winning positions with numbers swapped

We would like to see how that line compares with the line given by $(\lfloor (i-1)\phi \rfloor, \lfloor (i-1)\phi^2 \rfloor)$.

```
import numpy as np
import matplotlib.pyplot as plt
values = win_pos(100)
x = [values[pos][0] for pos in range(len(values)) ]
y = [values[pos][1] for pos in range(len(values)) ]

plt.plot(x, y, label="(m,n)")
plt.plot(y,x, label="(n,m)")
plt.legend()
plt.grid()
```

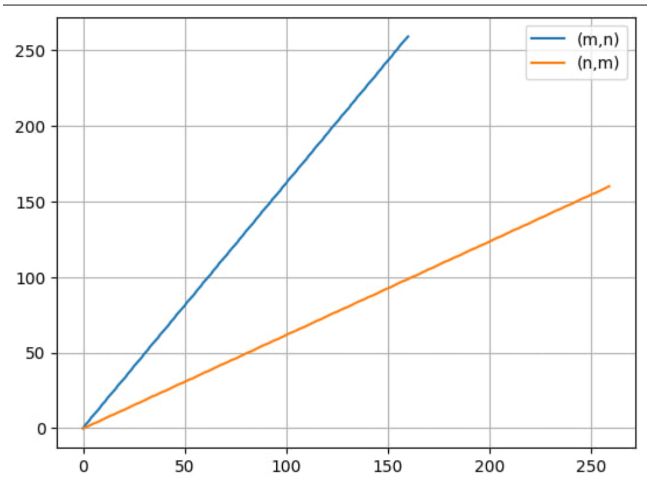

Figure 3: (n, m) plotted against (m, n)

So we see that $((\lfloor(i-1)\phi\rfloor, \lfloor(i-1)\phi^2\rfloor))$ has slope about $1.6180244133494199 \approx \phi$ and $(\lfloor(i-1)\phi^2\rfloor, \lfloor(i-1)\phi\rfloor)$ has slope about $0.6180244133494199 \approx \frac{1}{\phi}$.

Notice that in the graph of the winning positions, $(\lfloor(i-1)\phi^2\rfloor, \lfloor(i-1)\phi\rfloor)$ against $(\lfloor(i-1)\phi\rfloor, \lfloor(i-1)\phi^2\rfloor)$, the non-winning positions are the points (m, n) in the image above, with integer coordinates that are not on the lines for the winning positions and they fall into three categories:

1. Above the line with slope ϕ, that is, $\frac{n}{m} > \phi$
2. Below the line with slope $\frac{1}{\phi}$, that is, $\frac{n}{m} < \frac{1}{\phi}$
3. In between the two lines, that is, $\frac{1}{\phi} < \frac{n}{m} < \phi$

For any point that is in category 1 then it is always possible to move down to some winning position. For any point that is in category 2 it is always possible to move right to some winning position and for any point that is in category 3, it is always possible to move diagonally to some winning position.

So we formulate the strategy for the game as follows:
Given any two numbers (m, n) do the following:

1. Find the ratio $\frac{n}{m}$ and determine its relationship to ϕ (whether it is greater than, less than, or equal to ϕ)
2. If the ratio $\frac{n}{m}$ equals ϕ or it equals $\frac{1}{\phi}$ then we already have a winning position.
3. Based on that relationship, classify the points into one of the categories in the classification of non-winning positions above.
4. Now based on the categorization of the point, we determine from which side to subtract from (if the point is in category 1, above then we subtract from the right side only, if the point is in category 2 above, we subtract from the left side only, and if the point is in category 3 above, we subtract from both piles)

5. After determining which side to subtract from, you can now determine how much to move
   (a) To move down (subtracting from the right side) you calculate the winning positions until you get one with one of its coordinates equal to the coordinate on the left hand side of your position.
   (b) Similarly, to move right (subtracting from the left side) you calculate the winning positions until you get one with one its coordinates equal to the coordinate on the right hand side of your position.
   (c) Final to move diagonally (subtracting from both piles) you calculate winning positions until you get one whose coordinates are closest to, but less than your position. For each of those winning positions (x_i, y_i) for $i \in \{1, 2, 3\}$ compute $m-x_i$ and $n-y_i$ until you get $m - x_i = n - y_i$ for some i.
6. Finally subtract the appropriate amount from the appropriate side of your given position.

Below is a python-based implementation for this strategy.

```
import numpy as np
def find_win_pos(m,n):
  if m>n:
    m,n = n,m

  golden_ratio = (1+np.sqrt(5))/2
  golden_ratio_reci = 1/golden_ratio
  num_ratio = n/m
  N = max([m,n])+5
  size = int(np.ceil(N/golden_ratio**2))
  print(size)

  winning_positions = win_pos(size)
  possibilities = []


  if num_ratio > golden_ratio:
    for x,y in winning_positions:
      if x==m:
        possibilities.append((m,y))
      elif y==m:
        possibilities.append((m,x))
  elif num_ratio < golden_ratio_reci:
    for x,y in winning_positions:
      if y==n:
        possibilities.append((x,n))
      elif x==n:
        possibilities.append((y,n))
  if golden_ratio_reci < num_ratio and num_ratio < golden_ratio:
    for x,y in winning_positions:
      if m-x == n-y:
```

```
      possibilities.append((x,y))
elif num_ratio == golden_ratio or num_ratio == golden_ratio_reci:
  return (m,n)
return max(possibilities)
```