MATLAB Project: Card Game and Project Euler Problems

Name: Sicheng Wang, Jiannan Tang, Ariel Yao

This MATLAB Live Script contains one medium problem and three Project Euler problems.

Medium Problem: The Card Game

Project Euler Problems: Problem 31, Problem 36, and Problem 76

Part 1: Medium Problem — The Card Game

Problem:

A standard 52-card deck has 26 black cards and 26 red cards. A black card gives +$1 and a red card gives -$1. The player may stop at any time. The goal is to find the expected gain under optimal play and describe the optimal strategy.

Strategy:

I use dynamic programming. The state is determined by the number of black cards and red cards remaining. Let V(b,r) be the optimal expected future gain when b black cards and r red cards remain. At each state, the player compares stopping immediately, which gives value 0, with drawing one more card. This follows the principle of optimality from dynamic programming, where the optimal future decision depends only on the current state and the remaining subproblem.

Card Game Dynamic Programming

Source: MIT OpenCourseWare 6.231 Lecture 2, principle of optimality.

URL: https://ocw.mit.edu/courses/6-231-dynamic-programming-and-stochastic-control-fall-2015/resources/mit6_231f15_lec2/

```
clear; clc; close all;
format long g;

n = 26;

% V(b+1,r+1) = optimal expected future gain when b black cards
% and r red cards remain.
V = zeros(n+1, n+1);
Draw = false(n+1, n+1);

for total = 1:2*n
    for b = 0:min(n,total)
        r = total - b;

        if r < 0 || r > n
            continue;
```

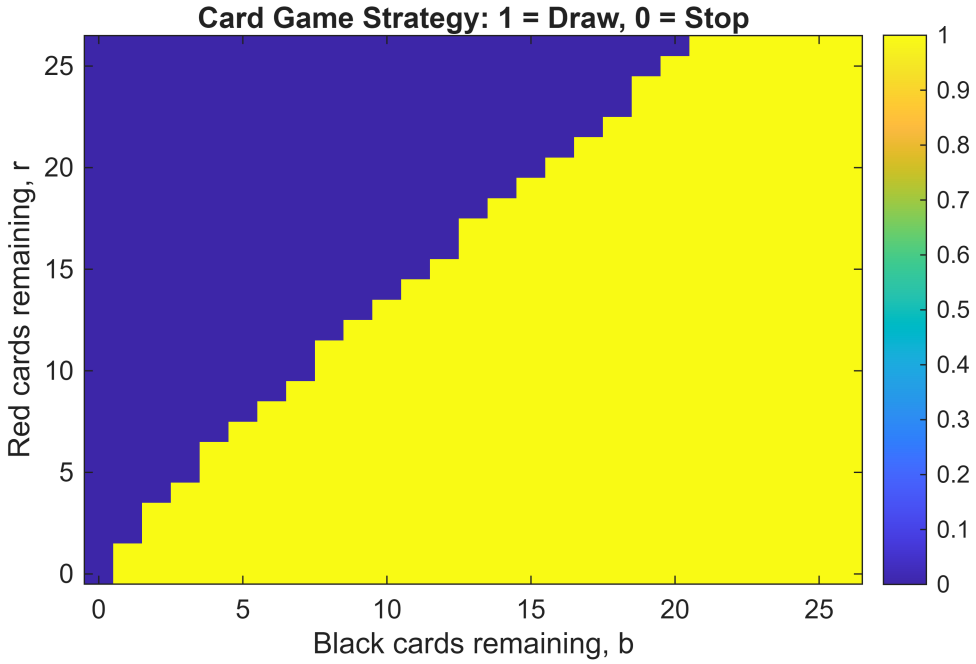
```
        end

        if b == 0
            % Only red cards remain, so stopping is optimal.
            V(b+1,r+1) = 0;
            Draw(b+1,r+1) = false;

        elseif r == 0
            % Only black cards remain, so drawing all remaining black cards is
optimal.
            V(b+1,r+1) = b;
            Draw(b+1,r+1) = true;

        else
            drawValue = (b/(b+r)) * (1 + V(b,r+1)) + ...
                        (r/(b+r)) * (-1 + V(b+1,r));

            V(b+1,r+1) = max(0, drawValue);
            Draw(b+1,r+1) = drawValue > 0;
        end
    end
end

cardAnswer = V(n+1,n+1);

fprintf("Expected gain for %d black and %d red cards: %.12f\n", n, n, cardAnswer);
```

```
Expected gain for 26 black and 26 red cards: 2.624475548994
```

```
fprintf("Check for 1 black and 1 red card: %.12f\n", V(2,2));
```

```
Check for 1 black and 1 red card: 0.500000000000
```

```
figure("Name", "Card Game Optimal Strategy");
imagesc(0:n, 0:n, Draw');
axis xy;
xlabel("Black cards remaining, b");
ylabel("Red cards remaining, r");
title("Card Game Strategy: 1 = Draw, 0 = Stop");
colorbar;
```



Result:

The expected gain under optimal play is approximately 2.624475548994 dollars. The 2-card check gives 0.500000000000, matching the example in the problem statement.

Optimal strategy:

At state (b,r), draw if the expected value of drawing is positive. Otherwise, stop.

Citation:

Source: MIT OpenCourseWare, 6.231 Dynamic Programming and Stochastic Control, Lecture 2: “The Basic Problem, Principle of Optimality, The General Dynamic Programming Algorithm, State Augmentation.”

URL: https://ocw.mit.edu/courses/6-231-dynamic-programming-and-stochastic-control-fall-2015/resources/mit6_231f15_lec2/

Part 2: Project Euler Problem

Project Euler

Problem 31: Coin Sums

In the United Kingdom the currency is made up of pound (£) and pence (p). There are eight coins in general circulation:1p, 2p, 5p, 10p, 20p, 50p, £1 (100p), and £2 (200p).

It is possible to make £2 in the following way:

1×£1 + 1×50p + 2×20p + 1×5p + 1×2p + 3×1p

How many different ways can £2 be made using any number of coins?

Strategy:

We use dynamic programming. The array ways stores how many combinations can make each amount.

For each coin, we update the number of ways to make all amounts up to 200. This avoids counting the same combination in different orders.

```
coins = [1 2 5 10 20 50 100 200];
target = 200;
ways = zeros(1, target + 1);
ways(1) = 1;
for coin = coins
    for amount = coin:target
        ways(amount + 1) = ways(amount + 1) + ways(amount - coin + 1);
    end
end
answer = ways(target + 1);
disp(answer)
```

```
       73682
```

Result:

The number of ways to make £2 is 73682.

Citation:

Source: Project Euler Problem 31, “Coin Sums.”

URL: https://projecteuler.net/problem=31

AI Use: ChatGPT was used to check correctness

Problem 36 — Double-base Palindromes

The decimal number, 585=1001001001 2 (binary), is palindromic in both bases.

Find the sum of all numbers, less than one million, which are palindromic in base 10 and base 2.

Strategy:

We check every number below 1,000,000. For each number, we convert it into a decimal string and a binary string.

A number is valid if both representations are palindromes. We sum all such numbers.

```
total = 0;
for n = 1:999999
    decimalStr = num2str(n);
```

```
    binaryStr = dec2bin(n);
    if strcmp(decimalStr, reverse(decimalStr)) && strcmp(binaryStr,
reverse(binaryStr))
        total = total + n;
    end
end
answer36 = total
```

```
answer36 =
      872187
```

Result:

The sum of all numbers below one million that are palindromic in base 10 and base 2 is 872187.

Citation:

Source: Project Euler Problem 36, “Double-base Palindromes.”

URL: https://projecteuler.net/problem=36

AI Use: ChatGPT was used to check correctness

Problem 76 — Counting Summations

It is possible to write five as a sum in exactly six different ways:

$$4+1$$
$$3+2$$
$$3+1+1$$
$$2+2+1$$
$$2+1+1+1$$
$$1+1+1+1+1$$

How many different ways can one hundred be written as a sum of at least two positive integers?

Strategy:

This is a partition problem. We use dynamic programming similar to Problem 31.

We count the number of ways to write 100 as a sum of positive integers.

We use numbers 1 through 99 (excluding 100 itself) to ensure at least two terms.

```
target = 100;
parts = 1:99;
ways = zeros(1, target + 1);
ways(1) = 1;
for part = parts
    for amount = part:target
        ways(amount + 1) = ways(amount + 1) + ways(amount - part + 1);
    end
end
answer76 = ways(target + 1)
```

```
answer76 =
   190569291
```

Result:

The number of ways to write 100 as a sum of at least two positive integers is 190569291.

Citation:

Source: Project Euler Problem 76, “Counting Summations.”

URL: https://projecteuler.net/problem=76

AI Use: ChatGPT was used to check correctness

References

Project Euler. “Problem 31: Coin Sums.” https://projecteuler.net/problem=31

Project Euler. “Problem 36: Double-base Palindromes.” https://projecteuler.net/problem=36

Project Euler. “Problem 76: Counting Summations.” https://projecteuler.net/problem=76

Bertsekas, Dimitri P. “6.231 Fall 2015 Lecture 2: The Basic Problem, Principle of Optimality, The General Dynamic Programming Algorithm, State Augmentation.” MIT OpenCourseWare. https://ocw.mit.edu/courses/6-231-dynamic-programming-and-stochastic-control-fall-2015/resources/mit6_231f15_lec2/

The MathWorks Inc. MATLAB. The MathWorks Inc., Natick, Massachusetts. https://www.mathworks.com

OpenAI ChatGPT. Used for code checking