UCB Math 124, Spring 2023: Final Exam

Prof. Persson, May 11, 2023

Name: ______________________________

SID: ______________________________

Instructions:

- One sheet of notes, no books, no calculators.
- Exam time 180 minutes, do all of the problems.
- You must justify your answers for full credit.
- All computer codes must be written in the Julia programming language, using only the functionality and the packages covered in the course.
- Write your answers next to or below each problem.
- If you need more space, use reverse side or scratch pages.
  Indicate clearly where to find your answers.

1. (6 points) Find the asymptotic operation counts and memory usage for the Julia functions below, using Big-O notation with brief justifications.

a)

```
x = rand(n)
s = 0.0
for i = 1:n, j = i+1:n
    s += abs(x[j] - x[i])
end
```

b)

```
s = 0.0
for i = 1:n, j = 2:2:n
    k = n
    while k > 0
        k ÷= 3
        s += 1 / (i + j + k)
    end
end
```

c)

```
x = Float64[]
for i = 1:n
    for j = 1:n
        if j > 1000
            break
        end
        newx = randn()
        if j % 10 == 0
            push!(x, newx)
        end
    end
end
```

2. Consider the following game: Given an integer N, random integers $x \in 1, 2, \ldots, N-1$ are drawn (with replacement) until the sum of any two numbers equals N. The "score" is the number of drawn numbers D. For example, if $N = 20$ and you get these numbers:

$$5, 19, 9, 8, 3, 12$$

then the score $D = 6$ since $8 + 12 = 20$.

a) (3 points) Write a Julia function `sumgame(N)` which simulates one instance of the game and returns the score D. The operation count for your code must be $\mathcal{O}(D)$ and the memory usage $\mathcal{O}(N)$. *Hint*: Use a boolean array to record the numbers.

b) (1 points) Write a Julia function `average_score(N,ntrials)` which uses Monte Carlo simulation with `ntrials` trials to estimate the expected value of the score.

3. If p is the perimeter of a right angle triangle with integer length sides, $\{a, b, c\}$, with $a \leq b \leq c$, there are exactly $n = 3$ solutions for $p = 120$:

$$\{20, 48, 52\}, \{24, 45, 51\}, \{30, 40, 50\}$$

a) (3 points) Write a Julia function `integer_triangles(p)` which returns the number of solutions n for the perimeter p, using operation count $\mathcal{O}(p^2)$ and memory $\mathcal{O}(1)$.

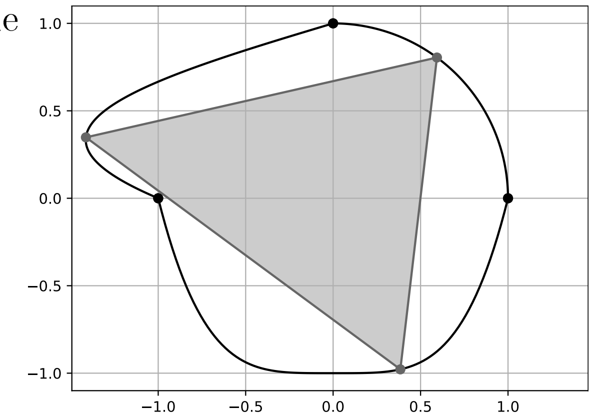
b) (1 points) Write a *one-line* Julia code which finds the value of $p \leq 1000$ for which n is maximized (use `argmax` to find the index of the largest element of an array).

4. (4 points) Consider the n-by-n matrix defined as follows, where n is an odd integer: Start in the central column of the first row with the number 1. Next, insert increasing numbers by moving diagonally up and right one step at a time. When an "up and to the right" move would leave the square, it is wrapped around to the last row or first column, respectively. If a filled square is encountered, move vertically down one square instead, again wrapping around, then continue as before. See below for an example when $n = 5$.

$$\begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix}$$

Write a Julia function `mysquare(n)` which creates and returns this matrix for size n.

5. (4 points) The shape in the picture is composed of the following three parametric functions:



$$(x, y) = (\cos t, \sin t), \quad 0 \le t \le \pi/2$$
$$(x, y) = (-\sqrt{2}\sin(3\pi t/4), 1 - t), \quad 0 \le t \le 1$$
$$(x, y) = (t, t^4 - 1), \quad -1 \le t \le 1$$

Consider a triangle with one vertex on each of these three curves. Use the `Optim` package, with the `Newton()` solver and the `autodiff=:forward` option, to find the triangle with the maximum area. Use the midpoints of the parameter range for each curve as initial guesses. You can assume that the parameters will remain within their ranges. Implement your code in a function `max_tri_area()` which returns a vector with the three x, y-coordinates of the optimal triangle.