Please refer to the corresponding exercise sections in the textbook (Rosen, 7th edition). Justify all answers in order to receive full credit.

1. Expansion Method (2 points)

For the recurrence below, use the expansion method to find a guess of what it could solve to. Make your guess as tight as possible.

\[ T(1) = 1, T(n) = 2T(n/2) + n \text{ for } n \geq 2 \]

2. Master Method (8 points)

Use the Master Theorem to solve the following recurrences. Justify your answers.

\begin{align*}
(1) & \quad T(n) = 3T(n/3) + 1 \\
(2) & \quad T(n) = 27T(n/3) + n^5 \\
(3) & \quad T(n) = 25T(n/5) + 5n^2 \\
(4) & \quad T(n) = 8T(n/2) + n^2
\end{align*}

3. Recursive Program (6 points)

Consider the following recursive function for \( n \geq 1 \):

```
Algorithm 1 int recurseFunc(int n)
    If n == 0, return 1.
    If n == 1, return 1.
    i = 0
    while i < n do
        j = 0
        while j < n do
            print(“hi”)
            j = j + 1
        end while
        i = i + 1
    end while
    int a = recurseFunc(n/9);
    int b = recurseFunc(n/9);
    int c = recurseFunc(n/9);
    return a + b + c
```

Continued on the back →
(1) Set up a runtime recurrence for the runtime $T(n)$ of this algorithm.
(2) Solve this runtime recurrence using the master theorem.

4. Linear Recurrence Relations (6 points)

Section 8.2 problem 4, parts a and b. Use the theorem we covered in class.