1. (7 points) Consider the following pseudocode for Dijkstra’s Algorithm:

```cpp
Dijkstra(graph G, vertex sourceVertex)
{
    bool finalized[G.nbrOfVertices] = { false };
    int minCost[G.nbrOfVertices] = { INFINITY };
    minCost[index(sourceVertex)] = 0;
    while ( finalized[index(v)] == false for some vertex v )
    {
        loop through vertices, find unfinalized vertex u w/smallest minimum cost;
        finalized[index(u)] = true;
        for every unfinished vertex w that is adjacent to u
        {  
            if (minCost[index(w)] > minCost[index(u)] + edgeCost(index(u), index(w)))
            {
                minCost[index(w)] = minCost[index(u)] + edgeCost(index(u), index(w));
                predecessor[index(w)] = index(u);
            }
        }
    }
}
```

Assuming that the number of available processors is less than the number of vertices in the graph, explain how to parallelize Dijkstra’s Algorithm to balance the load between multiple processors. Specify which steps in the above algorithm can be parallelized. Also, specify which data can be distributed to the different processors, as well as which data must be shared between all of them.

2. (2 points each) Image processing algorithms often rely on neighboring pixels. For instance, when blurring a grayscale image, the new pixel value could be the average of the values in its eight neighboring pixels, as illustrated at right.

Examine how to apply the PCAM method to these types of problems by explaining each of the following steps for the blurring problem. Assume that there are 16 processors and that the image being blurred has n x n pixels, where n is evenly divisible by 16.

a. Partitioning: Explain how to effectively partition the algorithm into individual units of computation.

b. Communication: Explain what communication will be required between the various partitioned units.

c. Agglomeration: Explain how to assign the partitioned units to disjoint groups between which communication is minimized.

d. Mapping: Assuming that communication and data sharing are equally efficient across all processors, explain how to assign the agglomerated groups to the processors.

3. EXTRA CREDIT (10 points) In a multiprocessor system, illustrated at right, several separate physical processors are installed on a computer, allowing multiple threads or processes to run simultaneously in the machine.

Multicore systems, on the other hand, take advantage of advances in integrated circuit manufacturing to place
multiple execution cores within a single processor chip. Depending on the design, these cores may or may not share a large on-chip cache, as illustrated in the two diagrams below.

For each of these three types of systems (multiprocessor, multicore without shared cache, and multicore with shared cache), explain the pros and cons of using the system with both SIMD and MIMD algorithms.

You must provide your own solutions to these problems in a clearly presented Word document. Obtaining solutions from any outside source is considered academic misconduct. The only person with whom you may discuss these problems is the instructor.