Quiz 5 answers – Aditya Shankar 5454360

Q1. Problem size is the number of basic computational steps in the best sequential algorithm for a particular problem. For example, in matrix-vector computation, where the matrix is of size (m x n), and the vector is of size (n x 1), the number of basic operations is of order O(m*n), i.e., m row computations with n iterations each, which is the problem size.

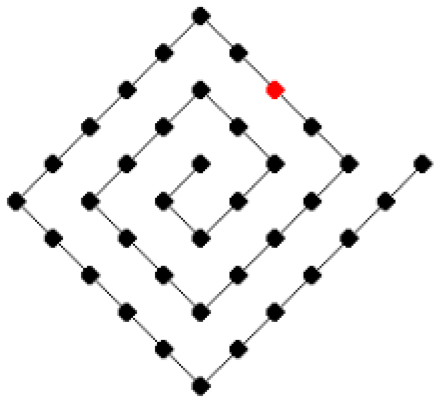
T_0 is the overhead time between a sequential program and a corresponding parallel implementation. If there are P processors and the corresponding sequential and parallel execution times are T_S and T_P respectively, then the overhead is PT_P - T_S. In the ideal case, this should be zero. This overhead can be due to multiple factors: load imbalance, communication with other processors, memory access delays, synchronization, coordination between other processors, etc.

Q2. Graph partitioning methods are broadly divided into two categories:

1) Using nodal coordinates

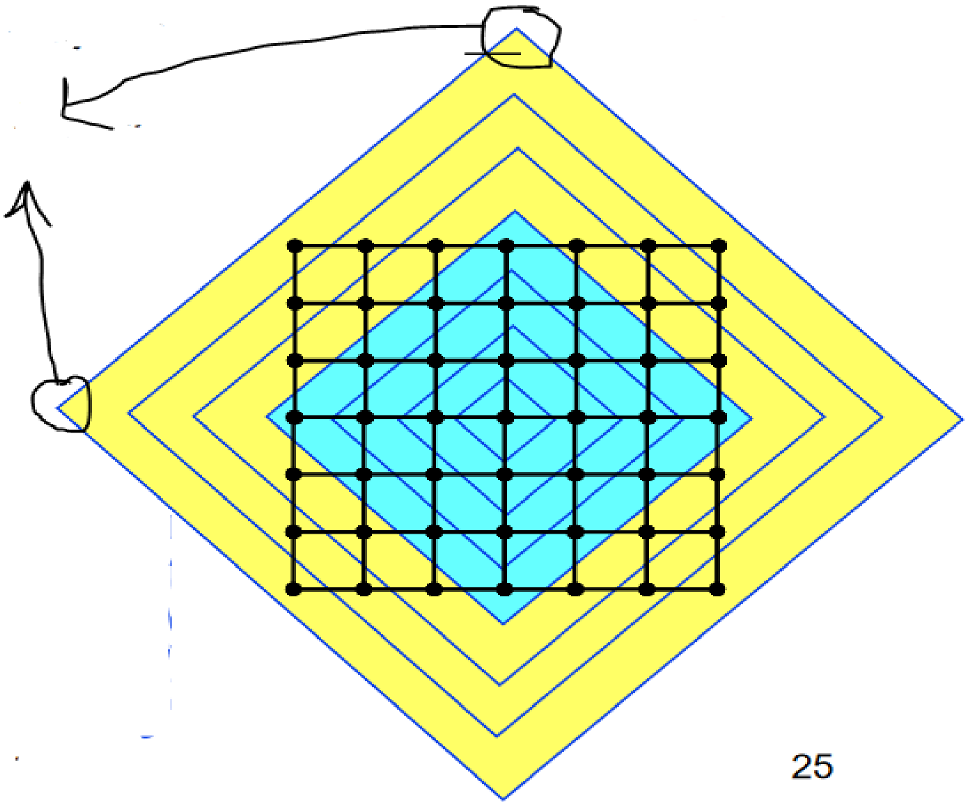
2) Coordinate-free partition algorithms.

Nodal coordinate algorithms perform a partition of the graph by “mapping” the grid points onto a coordinate system and then partitioning based on locality constraints. For example, the **Inertial Partitioning** algorithm partitions using a least-squares fit line followed by another perpendicular line that divides the points into approximately equal halves. This algorithm however does not work well if the graph is not spatial. For example, in the figure below, the optimal cut-point is the red point that divides the 1-D graph into equal halves, but since this graph has been folded, the inertial partitioning algorithm does not pick this point and hence results in a non-optimal cut-point.

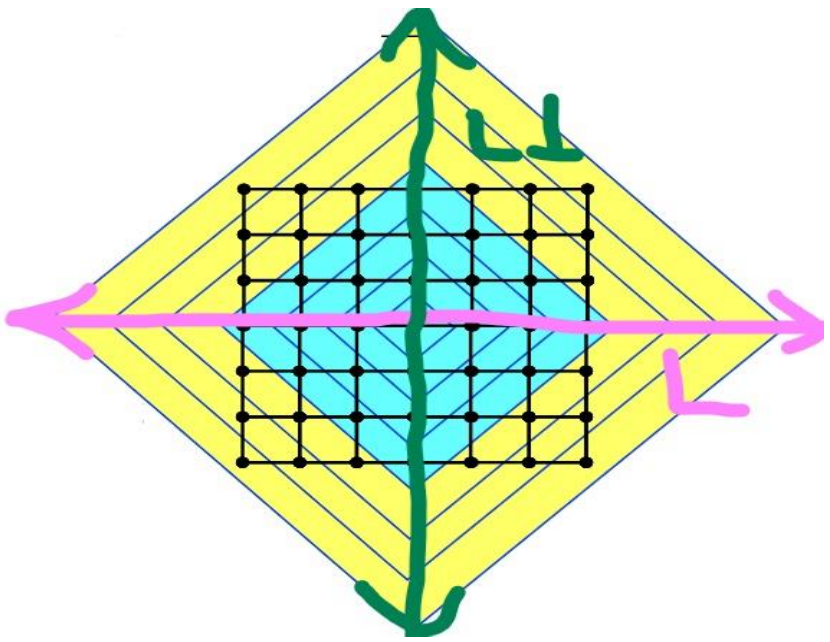


BFS (Breadth-First search algorithm avoids this issue as it does not use a coordinate-system. Here however, a root node must be selected and a tree is constructed using breadth-first-search. It is designed in such a way that nodes at the same level are included as part of the same partition. The drawback of this method is that a poor choice of the root node can give bad results. For example, in the figure below, a good choice for the root node would be one of the corner points. However, by choosing the centre as the root node, there are six levels formed, and in the last level we see that nodes that are quite far apart are assigned to the same partition. (Loss of locality). This is bad because it leads to higher communication costs within the partition.

Same level but very far apart, so partition is not good



Inertial partitioning on the other hand would give a better result in this case because it would form a division at approximately the middle of the grid. Hence locality is preserved (see fig below).



Hence BFS is not always better than Inertial partitioning. The choice depends on whether the graph is spatial (for inertial) and on the root node (in case of BFS).

Q3. The operations performed are: 3 multiplications, 1 division, and 2 additions. The memory accesses are 3: one for pixel. R, pixel. G, and pixel. B, each. Since each of these has a size of 4 bytes, the total memory accessed is 4 * 3 = 12 bytes. So, the operational intensity is: **(3 + 1 + 2) / (3 * 4) = 6/12** = **0.5.** Since this ratio is lesser than one, it means that the kernel spends longer time on memory access rather than in computation, so it is a **memory-intensive** kernel.