|  |  |  |
| --- | --- | --- |
| 1. | True/False | |
|  | Q: | T/F: Because matrices are square (n2), matrix multiplication is a O(n2) operation. |
|  |  |  |
|  | A: | False, it is n3, for obvious reasons. |
|  |  |  |
|  | Ref: | Page 342 |

|  |  |  |
| --- | --- | --- |
| 2. | Multiple Choice | |
|  | Q: | Which of the following is not a method of matrix multiplication discussed in the text:   1. mesh 2. recursive 3. SUMMA 4. direct |
|  |  |  |
|  | A: | C. SUMMA was not explicitly discussed in the text. |
|  |  |  |
|  | Ref: | Page 343-349 |

|  |  |  |
| --- | --- | --- |
| 3. | Fill in the Blank | |
|  | Q: | Using n3 processors is \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ optimal for matrix multiplication because the cost of sequential multiplication is O(n3) < n3\*O(lg n), the cost of matrix multiplication with n3 processors. |
|  |  |  |
|  | A: | not; cost. Such a setup for matrix multiplication is not cost optimal. |
|  |  |  |
|  | Ref: | Page 343 |

|  |  |  |
| --- | --- | --- |
| 4. | Short Answer/Code | |
|  | Q: | Briefly describe an advantage of Cannon’s algorithm over other matrix multiplication algorithms that do not use a sub-matrix scheme. |
|  |  |  |
|  | A: | Cannon’s algorithm allows for matrices that are too large to fit on one “box” simultaneously to be multiplied, and then the result does not need to be gathered at the end (unless one absolutely needs it); the sub-matrices can be stored in a distributed fashion. |
|  |  |  |
|  | Ref: | Page 348 |