1. Modification in Sequential Algorithm

- We can observe that during modified multiplication dependency is only on the loop k since we are taking minimum over that loop
- But, Two outer loops are independent with respect to each other
- Hence, we can give one outer loop and one middle loop to the different processors
- Hence simply we are giving N/P iterations to each processor
- We are only parallelizing the modified_matrix_multiplication function here
- We will not be parallelizing the While loop in All_Pairs_Distance function since it needs to be run sequentially, also its only takes log(n) steps to complete

2. Pseudo Code

```
Modified_Matrix_Multiplication(A,B):

    C ← inf // matrix of len(A) × len(B[0])

    // This loop can run in parallel
    LOOPS (i, 0, len(A)):
        LOOPS (j, 0, len(B)):
            LOOPS (k, 0, len(B[0])):
                C[i][j] ← min(C[i][j], A[i][k] + B[k][j])
            END
        END
    END

    RETURN C
```

3. Time Complexity

Time complexity for Modified Matrix multiplication
= (N/p)*N*N
= (N ^ 3) / p

Time complexity for iterating till D(n) = log(n)

Overall time complexity : (N ^ 3) * log(n) / p