**2. (a) Suppose an algorithm A1 divides (in linear time) the input into six equal parts: P1, P2, P3, P4, P5, P6. It then makes the following 10 combinations of these parts: each odd part is combined with each even part: P1 with P2, P4, P6 and so on, and in addition P1 is combined with P3. The algorithm is then run recursively on each of those 10 combinations. The results of the recursive runs are combined in quadratic time to produce the final output.**

* T(n) = a T(n / b) + f(n)
* In this case - b = 3 (Since input is divided into 6 parts and 2 of those parts are combined), a = 10, f(n) is quadratic - (n^2)
  + T(n) = 10 T(n / 3) + (n^2)
* By Master Theorem – if f(n) = O(n ^ k) for some constant k < log(a base b), then
* T(n) = Θ(n ^ log(a base b))
* In our case k = 2, a = 10, b = 3, log (10 base 3) = 2.09 > k
* Therefore Finally, T(n) = Θ(n ^ log(10 base 3))

**(b)** **Another algorithm, A2, divides the input into nine equal parts and runs on a (some number) of them recursively. It then combines the results of the recursive runs in quadratic time to produce the final output.**

By Master Theorem - T(n) = a T(n / b) + f(n)

* In this case - b = 9, a = a, f(n) is quadratic - (n^2) Therefore,
  + T(n) = a T(n / 9) + (n^2)