Solution Q7

Part 1

To merge two sorted arrays with size n each using the merge algorithm in merge sort, we need to make minimum n comparisons in the best case and $2n - 1$ comparisons in the worst case. Let X be a random variable denoting the number of comparisons done in addition to the n comparisons. The sample space of $X = \{0, 1, 2, \dots, n-1\}$. Also, all these values are equally probable. Hence the probability of X taking a value x in the sample space is given by $p(x) = \frac{1}{n}$
The number of comparisons during a merge is $n + X$. Then the average number of comparisons is given by,

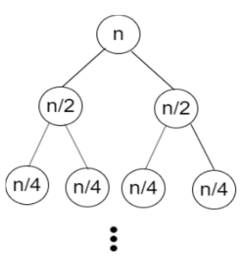
$E[n + X]$ = $E[n] + E[X]$ (linearity of expectation)

= $n + E[X]$ (n is a constant)

= $n + \sum_{x=0}^{n-1} x.p(x)$

= $n + \frac{1}{n}\sum_{x=0}^{n-1} x$ = $n + \frac{1}{n}\frac{(n-1)n}{2}$ = $1.5n - 0.5$

Part 2



The total number of comparisons is given by the sum of comparisons done in each level of the merge-tree. For merge sort, the average case tree is equivalent to the worst-case tree. Assume the root to be in level 0. Now it is easy to verify that at level i, the merge sort does 2^{i-1} different merges, each with two arrays of size $\frac{n}{2^i}$. For example, in level 2, we do 2 different merges, each with two arrays of size $\frac{n}{4}$

Using the result of part 1, we know that each merging of arrays with size n_i takes $1.5n_i - 0.5$ comparisons. Thus, the total number of comparisons in level i is given by $2^{i-1}\left(1.5\frac{n}{2^i} - 0.5\right)$

The overall number of comparisons is given by,

$$\sum_{i=1}^{\lceil \log_2 n \rceil} 2^{i-1}\left(1.5\frac{n}{2^i} - 0.5\right)$$

= $\sum_{i=1}^{\lceil \log_2 n \rceil}\left(1.5\frac{n}{2} - \frac{1}{2}2^{i-1}\right)$

= $0.75n\lceil \log_2 n \rceil - 0.5\left(2^{\lceil \log_2 n \rceil} - 1\right)$

= $0.75n\lceil \log_2 n \rceil - 0.5n + 0.5$