Average Case Analysis
(Fill in the table cells with execution times)

	InpType1			InpType2			InpType3			InpType4		
	n=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000
Ver1	737200.0	1080400.0	14197660.0	728200.0	1066340.0	14630280.0	824000.0	1210020.0	15727800.0	422840.0	4108240.0	450272540.0
Ver2	101560.0	1367820.0	17450400.0	102100.0	1346100.0	18987140.0	117680.0	1569840.0	18884680.0	498680.0	4163180.0	448661140.0
Ver3	717400.0	1079200.0	14241800.0	724800.0	1088720.0	14881320.0	833600.0	1250100.0	15360780.0	423320.0	4144660.0	447443720.0
Ver4	788800.0	1132220.0	14372740.0	884000.0	1666580.0	15164780.0	103440.0	1347540.0	16843540.0	453600.0	4224400.0	450962600.0

Comments:
(Write your detailed comments about the average case running times)
(Worst case results and comments are on the next page)

In all of these versions, the average case complexity is O(n*log(n)). We see that the execution time increases as the input size increases, as expected. When we compare the input types, we can see that type1 and type 2 are very close to each other, while type 3 is a little too much and type 4 is far more than the others.

When the number of duplicated elements increases, the performance of the algorithm decreases since it gets a more quadratic complexity than its normal O(n*log(n)). Since the input range decreases through InpType1 to InpType4, the execution time must increase. The table supports this.

When we compare the performances of the versions of the algorithm, no big difference is seen other than Ver2. Ver2 has a little more execution times than the others. The execution times of the Ver4 are also slightly bigger than the other two also. No difference between the versions was expected actually, and we concluded that these differences are caused by the operations that are executed to determine the pivot. In the Ver2, the random operation affects too much and in Ver4 the median of three operations affects slightly.

Since the pivot is selected as the first element in the classical deterministic algorithm, an extra operation does not affect the execution time in the average case for the selection of the pivot.

In version 3, the first element is selected as the pivot again, as in the pivot selection version 1, and no extra operation execution time is added to the execution time for the pivot selection, unlike in version 2 and version 4. Therefore, the execution times of version 1 and 3 are close to each other and smaller than version 2 and version 4.

Worst Case Analysis
(Fill in the table cells with execution times)

	InpType1			InpType2			InpType3			InpType4		
	n=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000	*n*=100	*n*=1000	*n*=10000
Ver1	201300.0	1576850.0	16871452.0	108400.0	924650.0	9860329.0	82700.0	465080.0	4360176.0	421600.0	4133740.0	45510638.0
Ver2	93500.0	1382500.0	16221100.0	96200.0	1278200.0	17382700.0	111000.0	1492800.0	18731300.0	474600.0	4173770.0	44688402.0
Ver3	70300.0	1124100.0	13737000.0	70300.0	1037300.0	16207200.0	79100.0	1136600.0	15873100.0	430800.0	4111940.0	45009857.0
Ver4	76000.0	926300.0	12443300.0	77700.0	1032000.0	12879500.0	95200.0	1316800.0	15080300.0	451200.0	4287350.0	45469299.0

Comments:
(Write your detailed comments about the worst case running times)

The average case complexity is $O(n^2)$ for the worst case of the quick sort normally. However when we apply the randomized concepts or choose the pivot according to the median of three rule, the complexity gets O(n*log(n)) and the execution times gets near to the average case values. This can be seen in the table when comparing the Ver1 and the other versions.

We see that the execution time increases as the input size increases, as expected. When we compare the input types, we can see that type1 and type 2 are very close to each other, while type 3 is a little too much and type 4 is far more than the others. These are similar to the average case values.

When the number of duplicated elements increases, the performance of the algorithm decreases since it gets a more quadratic complexity. Since the input range decreases through InpType1 to InpType4, the execution time must increase. The table supports this.

The case for the Ver2 and Ver4 (additional operations for choosing pivots) are the same as explained in the comments of the average case section.

While the worst case complexity is $O(n^2)$, the complexity of Ver3 is O(n*log(n)) because we shuffle the incoming input no matter what the input is in version 3. There is no difference between the average case and the worst case when we shuffle before the algorithm. Therefore, the execution time of version 3 is smaller than all other versions.

In the Ver2 and Ver4, the method to choosing the pivot provides a more even dividing throughout the algorithm. This is the reason they have O(n*log(n)) complexity and the statistics are similar to the average case statistics.