Question 6 and 7

Question 6

Table of Statistics of Binary Search Trees

	Minimum Height	Maximum Height	Average Height	Upper Bound Expected Height	Worst Case Upper Bound of Red Black Tree
n = 100	**9**	**15**	**12.16**	**19.931568569324178**	**13.316422965503591**
n = 1000	**17**	**27**	**20.97**	**29.897352853986263**	**19.934452517671986**
n = 10000	**26**	**36**	**30.01**	**39.863137138648355**	**26.57571328368109**
n = 100000	**36**	**48**	**39.63**	**49.82892142331043**	**33.21930980263018**

Question 7

The data that I obtained in the previous question is very close to what I would expect, given that these are statistics taken from 100 binary search trees each. It seems to be the case that when 100 binary search trees are considered, the height of a binary search tree with ***n*** nodes approaches ***O(log(n))*** height. This is consistent with the theoretically accepted average height of randomly built binary search trees.

It also seems that there is some variation in the values even though we have chosen to sample 100 trees at a time. Overall however, the values do not change much. After running the program many times, I have seen only a slight deviation between values. Thus, it seems appropriate to claim that the values obtained in the previous question are consistent with what one would expect.

The upper bound on the expected height of a random binary search tree also seems appropriate, given that it is within the range of the calculated values. It may differ from the calculated average height, but there may be reasons for this. For instance, the value represents the expected height of a randomly constructed tree for sufficiently large ***n***. It could be that in order to approach this value, greater ***n*** values need to be used. We are also limited by the range of integers values which could be stored in the binary search tree, as well as the fact that these numbers are generated by a pseudo-random number generator. Thus, while it may not be an exact match with our calculated values, it does provide a reasonable estimate of the upper bound on the height of a randomly constructed binary search tree.

The worst case upper bound on the maximum height of a red black tree is also as expected, as it is consistently less than the upper bound expected height of a random binary search tree. This is due to the fact that a red black tree is always close to being full. We would expect that a red black tree has a slightly better upper bound than a random binary search tree. This is further supported by the fact that the worst case upper bound of red black trees is often less than the calculated height for the binary search trees.