Pruning strategy – "Batch pruning with randomly chosen nodes"

<u>Code Summary</u>

1) Create a dictionary that stores the best pruned tree and accuracy

2) List all the inner nodes into the list, which gets shuffled each time when executing the "driver.py".

3) Partitioning the inner node list into nested lists with the specified number of elements "n".

4) Set the number of pruning iterations:

```
    If number of partitioned lists less than 10
        run the pruning iterations within the limit 10
    else
        to avoid unbounded iterations when numerous attributes is considered, limit the
pruning iterations by (length of partitioned list)/2
```

5) For each batch list in the partitioned list, run the prune_tree function and find the accuracy of the pruned tree on the test data.

6) With the multiple iterations of pruning the tree with batch inner_nodes, when accuracy increases, update the dictionary with the best pruned tree and the best accuracy obtained.

7) For the next subsequent iterations, reset to the original tree using deep copy method before sending it to the pruning method.

Analogy:

	Iris Dataset	**Tic-Tac-Toe Dataset**
Count of inner nodes	**7**	**71**
Accuracy before pruning	**93%**	**97%**
Accuracy after pruning	**97%**	**98%**