```
recursiveFibonacci result:832040
executionTime for recursiveFibonacci :12913000


fibMemorization result:832040
executionTime for fibMemorization :603400


fibTabulation result:832040
executionTime for fibTabulation :986200


myFibonacci result:832040
executionTime for myFibonacci :599000
```

Recursion is the most time consuming.

Memorization(top-down approach) takes less time than tabulation(bottom-up).

Both memorization and tabulation are faster and more efficient than normal and recursive approach.

	Tabulation	Memoization
State	State Transition relation is difficult to think	State transition relation is easy to think
Code	Code gets complicated when lot of conditions are required	Code is easy and less complicated
Speed	Fast, as we directly access previous states from the table	Slow due to lot of recursive calls and return statements
Subproblem solving	If all subproblems must be solved at least once, a bottom-up dynamic-programming algorithm usually outperforms a top-down memoized algorithm by a constant factor	If some subproblems in the subproblem space need not be solved at all, the memoized solution has the advantage of solving only those subproblems that are definitely required
Table Entries	In Tabulated version, starting from the first entry, all entries are filled one by one	Unlike the Tabulated version, all entries of the lookup table are not necessarily filled in Memoized version. The table is filled on demand.

- https://www.youtube.com/watch?v=ewXklhesC-w&list=PLSIpQf0NbcClDpWE58Y-oSJro_W3LO8Nb&index=2&ab_channel=JAVAAID-CodingInterviewPreparation
- https://www.geeksforgeeks.org/tabulation-vs-memoization/