フィボナッチ数列をDPで解く練習 int n番目の値を考える

1. 全探索を考える
2. メモ化を考える
3. 動的計画法（forループ版）を考える

-------------------------------------------------------------------------------------------------------

「全探索」

フィボナッチ数列の確認

1 1 2 3 5 8 13 21 …

一つ前と二つ前の値が必要

→i番目を求めるときはi -2とi-1を足したものをiとする

一つ前の値はさらに一つ前と二つ前の値が必要と、どんどん値が小さいほうへ下っていく

再帰の全探索版では求めたいｎ番目の一つ前と二つ前の値を再帰で求める

-------------------------------------------------------------------------------------------------------

「メモ化」

* 一つ前と二つ前を求めるということは全探索の木が出来る
* 二つ前の値は一つ前の値を求めるときに使う。

→それ以降の計算は全く一緒。なので、一度計算すれば分かる値がある

具体的には、再帰するときにもう一度nを指定するが、そのｎで一度計算すればその戻り値は全く一緒となる。だから、これをメモ化する。

すると一度計算した値はメモから抜き出せる

プログラム上ではmemo[i]が計算されていれば０以外の値になるのでそうなれば計算されていることとなる。（グローバル変数はゼロクリアされている）

一番わかりやすいのは、if分をブレークポイントで止めて流れを見てみるとわかりやすい。

-------------------------------------------------------------------------------------------------------

「動的計画法版（forループ）」

メモ化の方法では木で考えたときに最初に一番深くまで潜り上に戻る時、既に計算している値を足し合わせながら上がっていく。

→どうにかして前から計算できないかを考える。

i番目とi+1番目を足した結果がi+2となる。

i+2がnの時ループを終了すればそこで求まった値は答えとなるはず。

0番目を0とした時1番目は0+1=1となる

→０と１さえ決まればi-2 + i -1 = iで順番に決まっていく。