Project Euler

Wafik Aboualim

June 16, 2022

Problem 2

Naive algorithm: we iterate over n and check for even Fibonacci number and sum them. Given we use the optimal Fibonacci algorithm, the algorithm will take $\mathcal{O}(n \log(n))$ time.

Optimal solution: We recall an important fact about Fibonacci numbers. If we take every term in the Fibonacci sequence $\Phi(n) \pmod 2$, we get the following sequence:

$$0, 1, 1, 0, 1, 1, 0, 1, 1, ...$$

In other words, for every natural number n, we have the following:

$$\begin{aligned}\Phi(n) = &f(3n) \text{ is even } \textbf{and}\\ &f(3n+1) \text{ is odd } \textbf{and}\\ &f(3n+2) \text{ is odd.}\end{aligned}$$

Proof. We induct over n, n=0 is trivial since the first 3 numbers of Fibonacci sequence are 0,1,1. Now suppose the above holds for $n = k$, we need to prove that it holds for $n = k+1$. Now consider $\Phi(3(k+1))$:

$$\Phi(3(k+1)) = \Phi(3k+3) = \Phi(3k+2) + \Phi(3k+1)$$

Since the sum of two odd numbers is even, hence $\Phi(3(k+1))$ is even. In the same manner it is easy to prove that both $\Phi(3(k+2))$, $\Phi(3(k+3))$ are odd. □

Using that fact, we can only iterate on the multiples of 3 until n. Unfortunately, that reduces our run time only by a constant.