1 Recursive Formulae

Suppose that we want to compute $2^n \mod M$ for some numbers $n \geq 0$ and $M \geq 2$. $2^n$ may require a lot of digits to write down for large $n$, and we want to avoid that, since the end result is $\leq M$.

Our first attempt avoids multiplication and only uses addition modulo $M$. We use the fact that $2^n = 1 + 2^n$. We prefer algorithms with the smallest worst-case runtime.

function PowerOfTwo(n, M):

    if $n = 0$
        return 1
    return $(\text{PowerOfTwo}(n-1, M) + \text{PowerOfTwo}(n-1, M)) \mod M$

What is the runtime of the above algorithm?
- $O(n)$
- $O(2^n)$
- $O(\log(n))$

Now let us replace this algorithm with an iterative one that stores the results:

$n$   $B_n$ $B_{n-1}$ $A_n$
------- ------- ------- ------
0      0      0      0
1      1      1      1
2      2      3      3
3      4      7      11

Suppose that we want to compute $\text{Fibonacci}(n)$ modulo a desired number $M$, in time $O(\log n)$. As a challenge, try to use the following identity involving Fibonacci numbers and matrix multiplication, to come up with this $O(\log n)$ algorithm:

\[
\begin{pmatrix}
Fibonacci(n) \\
Fibonacci(n+1)
\end{pmatrix}
= 
\begin{pmatrix}
1 & 1 \\
1 & 0
\end{pmatrix}^n
\begin{pmatrix}
1 \\
1
\end{pmatrix}
\]

2 Shortest Paths

Suppose that we have a weighted graph with $n$ vertices and $m$ edges and no negative cycles (so shortest paths are well-defined). Suppose for the below questions, that our implementation of Djikstra uses red-black trees (and not Fibonacci heaps).

If $n = 2^{10}$, and we want to find the shortest path between some $u$ and $v$ algorithms which should we use? We prefer algorithms with the smallest worst-case runtime.

- Djikstra
- Bellman-Ford
- Floyd-Warshall

Two or more of the above algorithms are correct and have the smallest worst-case runtime.

Correct

What if all the edges have nonnegative weight?

- Djikstra
- Bellman-Ford
- Floyd-Warshall

Two or more of the above algorithms are correct and have the smallest worst-case runtime.

Correct

Suppose that we have a graph with $n = (\log(4))^3$ edges that all have nonnegative weights. Which algorithm should we use to find the shortest path between all pairs of vertices?

- Djikstra
- Bellman-Ford
- Floyd-Warshall

Two or more of the above algorithms are correct and have the smallest worst-case runtime.

Correct