1. a) $A^1 = A$

$A^N = (A^{\lfloor N/2 \rfloor})^2$ if N even and ≥ 2

$A^N = (A^{\lfloor N/2 \rfloor})^2 \cdot A$ if N odd and ≥ 2

$\Rightarrow T(N) \leq T(\lfloor \frac{N}{2} \rfloor) + 2$

↓ multiplying $A^{\lfloor N/2 \rfloor} \cdot A^{\lfloor N/2 \rfloor}$ and then multiplying by A

- we can half the problem at most $\log N$ times

$\Rightarrow$ in each step we do a constant number of operations and we have $\log N$ steps at most $\rightsquigarrow O(\log N)$

b) $\binom{N}{k} \leq 2^N$ so it can be written with N bits

$A := 2^N \Rightarrow (A+1)^N = \sum_{0 \leq k \leq N} \binom{N}{k} 2^{Nk}$

$(A+1)^N = \underbrace{\binom{N}{0}}_{} \underbrace{\binom{N}{1}}_{} \ldots \underbrace{\binom{N}{k}}_{} \ldots \underbrace{\binom{N}{N}}_{}$

$N+1$ blocks of N bits each

- want to "extract" the k-th block:

$\lfloor (A+1)^N : 2^{Nk} \rfloor = \lfloor \sum_{0 \leq R \leq N} \binom{N}{R} 2^{NR} : 2^{Nk} \rfloor$ → when $R < K$, these elements are lost in the rounding down (get smth. < 1 when dividing)

$= \binom{N}{k} + \binom{N}{k+1} 2^N + \binom{N}{k+2} \cdot 2^{2N} + \ldots + \binom{N}{N} 2^{N \cdot N}$

$\lfloor (A+1)^N : 2^{Nk} \rfloor \bmod 2^N = \binom{N}{k}$

$\Rightarrow T(N) = O(\log N) + 2$

↳ $\lfloor : 2^{Nk} \rfloor$, $\bmod 2^N$

c) $1! = 1$

$N! = \binom{N}{\frac{N}{2}} \left(\frac{N}{2}\right)! \left(\frac{N}{2}\right)!$ if N even and ≥ 2

$N! = \binom{N}{\frac{N-1}{2}} \left(\frac{N-1}{2}\right)! \left(\frac{N-1}{2}\right)! \left(\frac{N+1}{2}\right)$ if N odd and ≥ 2

$T(N) \leq T(\lfloor \frac{N}{2} \rfloor) + O(\log N) + O(1)$

↳ multiply together

⇒ we can half our problem at most $\log N$ times and at each level we need $O(\log N)$ time

⇒ $T(N) = O(\log^2 N)$

d) need: N prime $\Leftrightarrow N \nmid (N-1)!$

[⇒] if N is prime, then it obviously does not divide $(N-1)!$ because $N > 1, 2, \ldots, N-1$ and cannot be split into smaller factors that could divide some numbers from $1, 2, \ldots, N-1$

[⇐] we prove N not prime $\Rightarrow N \mid (N-1)!$

(equivalent to $N \nmid (N-1)! \Rightarrow N$ prime)

assume 1) $N = \prod P_i^{e_i}$ and there is more than one prime factor $\Rightarrow P_i^{e_i} < N \Rightarrow$ contained in $(N-1)!$ $\forall i$

$\Rightarrow (N-1)!$ is a multiple of N ✓

2) $N = P^e \Rightarrow P, P^2, \ldots, P^{e-1} < N \Rightarrow$ all of them are contained in $(N-1)! \Rightarrow P \cdot P^2 \cdot \ldots \cdot P^{e-1}$ is contained in $(N-1)! \Rightarrow P^e = N$ contained in $(N-1)!$ ✓

test whether N is prime:

- just check whether N divides $(N-1)!$, for instance whether $\lfloor (N-1)! : N \rfloor \cdot N = (N-1)!$
- for this, we need to compute $(N-1)!$ and the rest is done in constant time $\Rightarrow T(N) = O(\log^2 N)$

e) If $N \mid Q!$, then $N \mid R!$ for any $R \geq Q$ (we are just adding more factors) and $N \nmid 1!$

↪ find the smallest K s.t. $N \mid K!$ and $N \nmid (K-1)!$ using binary search
- $GCD(K, N)$ is a factor of N

runtime: $T(N) \leq O(\log N) \cdot O(\log^2 N) + O(\log N)$

($\log N$: binary search; $\log^2 N$: each time computing a factorial; $\log N$: GCD)

$$= O(\log^3 N)$$

f) We find a divisor K like in e) and then do factorizations of K and N/K.

$$T(N) \leq \underbrace{T(N/K)}_{\text{one subproblem}} + \underbrace{T(K)}_{\text{other}} + \underbrace{O(\log^3 N)}_{\text{time needed to divide into these two subproblems}}$$

- we can divide our problem in this way for at most $\log N$ times (K always ≥ 2) and for each division need $O(\log^3 N) \Rightarrow T(N) = O(\log^4 N)$

(2.) a) $n \log n \leq n \cdot 2^{\sqrt{\log n}}$

$\log n \leq 2^{\sqrt{\log n}}$

$\log \log n \leq \sqrt{\log n} \quad / ^2$

$(\log \log n)^2 \leq \log n$

$2 \log \log \log n \leq \log \log n$

$$\frac{\log \log \log n}{\log \log n} \leq \frac{1}{2}$$ ✓

because $\lim_{n \to \infty} \frac{\log(\log(\log n))}{\log(\log n)} = 0$