Zusammenfassung: Kryptographie

Folgende Themen sind nicht prüfungsrelevant:

Quadratische Reste

Untergruppe von quadratischen Resten:
$QR_n = \{[x] \in \mathbb{Z}_p^* \mid \text{es gibt ein } [y] \in \mathbb{Z}_p^* \text{ mit } [x] = [y]^2\}$

Legendresymbol und Jakobisymbol

Legendresymbol: $\left(\frac{x}{\underbrace{p}_{prime}}\right) := \begin{cases} +1 & \text{wenn } [x] \in QR_p \\ -1 & \text{wenn } [x] \in QR_p \\ 0 & \text{wenn } p \mid x \end{cases}$

Jakobisymbol: $\left(\frac{x}{\underbrace{n}_{ungerade}}\right) := \prod_{i=1}^{r} \left(\frac{x}{p_i}\right)^{e_i}$

```
sbyte jac(uint64 x, uint64 n) {
    if(x == 0) {
        return n == 1 ? 1 : 0;
    } else if(x == 2) {
        switch(n % 8) {
        case 1: case 7:
            return 1;
        case 3: case 5:
            return -1;
        }
    } else if(x >= n) {
        return jac(x % n, n);
    } else if(x % 2 == 0) {
        return jac(2, n) * jac(x / 2, n);
    } else if(x % 4 == 3 && n % 4 == 3) {
        return -jac(n, x);
    } else {
        return jac(n, x);
    }
}
```

Modulare Quadratwurzeln

siehe Algorithmus

```
uint64 sqrtModulo(uint64 n, uint64 p) {
    if(!n) {
        return 0;
    } else if(p == 2) {
        return n % 2;
    } else if(p % 4 == 3) {
        return powerModulo(n, (p + 1) / 4, p);
    } else if(p % 8 == 5) {
        auto v = powerModulo(2 * n, (p - 5) / 8, p);
        return ( n * v * ( ((2 * n * v * v) % p) - 1 )  ) % p;
    } else {
        // TODO
    }
}
```