Problem 1.

Part A. Let $\mathbf{s}^b = (s_1^b, s_2^b, \ldots, s_n^b)$ be randomly sampled from $\mathsf{Share}(b, r)$, where $b \in \{0, 1\}$ and randomness comes from r.

By privacy for any $i \in [n]$,

$$\Pr[s_i^0 = s_i^1] = \sum_{s \in \mathcal{S}_i} \Pr[s_i^0 = s]^2 \geq \frac{1}{|\mathcal{S}_i|},$$

where the last inequality follows from Cauchy-Schwarz. $\sum_{i \in [n]} \frac{1}{|\mathcal{S}_i|} > 1$ will imply that

$$\sum_{i \in [n]} \Pr[s_i^0 = s_i^1] = \mathbb{E}[\#\{i : s_i^0 = s_i^1\}] > 1.$$

By probabilistic method, there exists $\mathbf{s}^0, \mathbf{s}^1$ meets in two coordinates, which contradicts with correctness.

Therefore apply Jensen's inequality, we have

$$-\log(1/n) \leq -\log\left(\sum_{i \in [n]} \frac{1}{n|\mathcal{S}_i|}\right) \leq \sum_{i \in [n]} -\frac{1}{n} \log\left(\frac{1}{|\mathcal{S}_i|}\right).$$

Part B. Condition on $s_1, s_2, \ldots, s_{t-2}$ fixed, $\{s_i : i \in [n], i \geq t-1\}$ forms 2-out-of-n secret sharing, by **Part A.**

$$\sum_{i \in [n], i \geq t-1} |\mathcal{S}_i| \geq (n - t + 2) \log(n - t + 2).$$

Problem 2.

Let $\mathbb{F}$ be a sufficiently large finite field and let $1, 2, \ldots, n, a_1, \ldots, a_{\ell-k}$ be distinct elements in $\mathbb{F}$.

The (k, ℓ, n)-ramp secret sharing is constructed as follows

- The secret space is $\mathbb{F}^{\ell-k}$.
- Given the secret $(s_1, \ldots, s_{\ell-k})$, the sharing algorithm samples a random P over $\mathbb{F}$ whose degree is at most $\ell - 1$, outputs $P(1), \ldots, P(n)$ as the shares.

Proof of correctness: Given any ℓ shares, the polynomial P can be recovered, thus the secret can be recovered.

Proof of privacy: No matter what the secret is, after fixing $P(a_1), \ldots, P(a_{\ell-k})$, there are still k degrees of freedom. The marginal distribution of any k shares is uniform.

Problem 3.

Say the two parties are Alice and Bob. Assume there exists such a 2-party computation protocol for AND that are perfectly secure against semi-honest adversary. W.l.o.g., we can assume two parties alternatively sends messages. I.e., Alice sends the first message, Bob sends the second message, Alice sends the third message, and so on.

Alice's first message should reveal nothing about Alice's input. Consider the case when Bob's input is 0, then the output must be 0, so Bob's view (including the first message) can be perfectly simulated without information about Alice's input.

Therefore, there is no need to send the first message. Just choose any first message form its support. (Alice will need to conditionally sample her state based on the first message and her input).

By the same argument, the second message is also useless. Recursively, all the communication are useless. Alice and Bob can compute AND function without any communication, which is impossible.

Problem 4.

Use the following protocol to compute a function f

- Let $(\mathsf{Gen}, \mathsf{MAC}, \mathsf{Verify})$ be a one-time MAC scheme. The i-th party samples key $k_i \leftarrow \mathsf{Gen}(1^\lambda)$.
- Use a PKO secure protocol to compute f', which is defined as

  $$f'((x_1, k_1), \ldots, (x_n, k_n)) = ((y_1, \mathsf{MAC}(k_1, y_1), \ldots, (y_n, \mathsf{MAC}(k_n, y_n)),$$

  where $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$. By assumption, such PKO secure protocol exists, denote the protocol by Π'. All honest parties learn $(\hat{y}, \hat{t})$ from Π'.
- The i-th party outputs $\hat{y}$ if $\mathsf{Verify}(k_i, \hat{y}, \hat{t})$ accepts. Otherwise, the i-th party aborts (i.e., outputs $\perp$).