Algorithm 1: Soft Actor-Critic (SAC)

Input: Initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, target Q-function parameters $\bar{\phi}_1 \leftarrow \phi_1$, $\bar{\phi}_2 \leftarrow \phi_2$, temperature parameter α, polyak averaging coefficient τ, empty replay buffer $\mathcal{D}$

for *each iteration* **do**
  Reset environment and observe initial state s_0;
  while *not terminal* **do**
    Sample action $a_t \sim \pi_\theta(a_t|s_t)$;
    Execute a_t in environment;
    Observe reward r_t, next state s_{t+1}, and terminal signal d_t;
    Store $(s_t, a_t, r_t, s_{t+1}, d_t)$ in replay buffer $\mathcal{D}$;
    if d_t *is True* **then**
      Reset environment and observe new initial state s_{t+1};
    end
    $s_t \leftarrow s_{t+1}$;
  end
  for *each gradient step* **do**
    Sample a mini-batch of N transitions (s, a, r, s', d) from $\mathcal{D}$;
    Compute target Q-value:

$$y = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\bar{\phi}_i}(s', a') - \alpha \log \pi_\theta(a'|s')\right), \quad a' \sim \pi_\theta(\cdot|s')$$

    Update Q-functions by gradient descent:

$$\phi_i \leftarrow \phi_i - \lambda_Q \nabla_{\phi_i} \frac{1}{N} \sum \left(Q_{\phi_i}(s, a) - y\right)^2, \quad i = 1, 2$$

    Update policy by gradient ascent:

$$\theta \leftarrow \theta - \lambda_\pi \nabla_\theta \frac{1}{N} \sum \left(\alpha \log \pi_\theta(a|s) - \min_{i=1,2} Q_{\phi_i}(s, a)\right), \quad a \sim \pi_\theta(\cdot|s)$$

    Update temperature:

$$\alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha \frac{1}{N} \sum \left(-\alpha \log \pi_\theta(a|s) - \alpha \mathcal{H}\right)$$

    Update target Q-functions:

$$\bar{\phi}_i \leftarrow \tau \phi_i + (1-\tau)\bar{\phi}_i, \quad i = 1, 2$$

  end
end