Potential Method

and amortized cost
example: binary counter

Amortized Cost

- (op_i) sequence of operations
- c_i cost of operation op_i

$$\sum_{i=1}^{n} c_i = ?$$

- Φ_i potential after operation op_i.
  An account. Reduce it to pay for expensive operations, fill during cheap operations.
- $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$ amortized cost

Amortized Cost

- (op_i) sequence of operations
- c_i cost of operation op_i

$$\sum_{i=1}^{n} c_i = ?$$

- Φ_i potential after operation op_i.
  An account. Reduce it to pay for expensive operations, fill during cheap operations.
- $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$ amortized cost

$$\begin{aligned}
\sum_{i=1}^{n} \hat{c}_i &= \sum_{i=1}^{n} (c_i + \Phi_i - \Phi_{i-1}) \\
&= \sum_{i=1}^{n} c_i + \Phi_n - \Phi_0 \\
\sum_{i=1}^{n} c_i &= \sum_{i=1}^{n} \hat{c}_i + \Phi_0 - \Phi_n
\end{aligned}$$

Amortized Cost

- (op_i) sequence of operations
- c_i cost of operation op_i

$$\sum_{i=1}^{n} c_i = ?$$

- Φ_i potential after operation op_i.
  An account. Reduce it to pay for expensive operations, fill during cheap operations.
- $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$ amortized cost

$$\begin{aligned}
\sum_{i=1}^{n} \hat{c}_i &= \sum_{i=1}^{n} (c_i + \Phi_i - \Phi_{i-1}) \\
&= \sum_{i=1}^{n} c_i + \Phi_n - \Phi_0 \\
\sum_{i=1}^{n} c_i &= \sum_{i=1}^{n} \hat{c}_i + \Phi_0 - \Phi_n
\end{aligned}$$

$$\boxed{\Phi_0 = 0 \,,\ \Phi_n \geq 0 \ \rightarrow\ \sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i}$$

example: binary counter

- binary counter with k bits
- op_i: increments counter by 1
- c_i: number of bits changed
- t_i: number of trailing ones after operation op_i

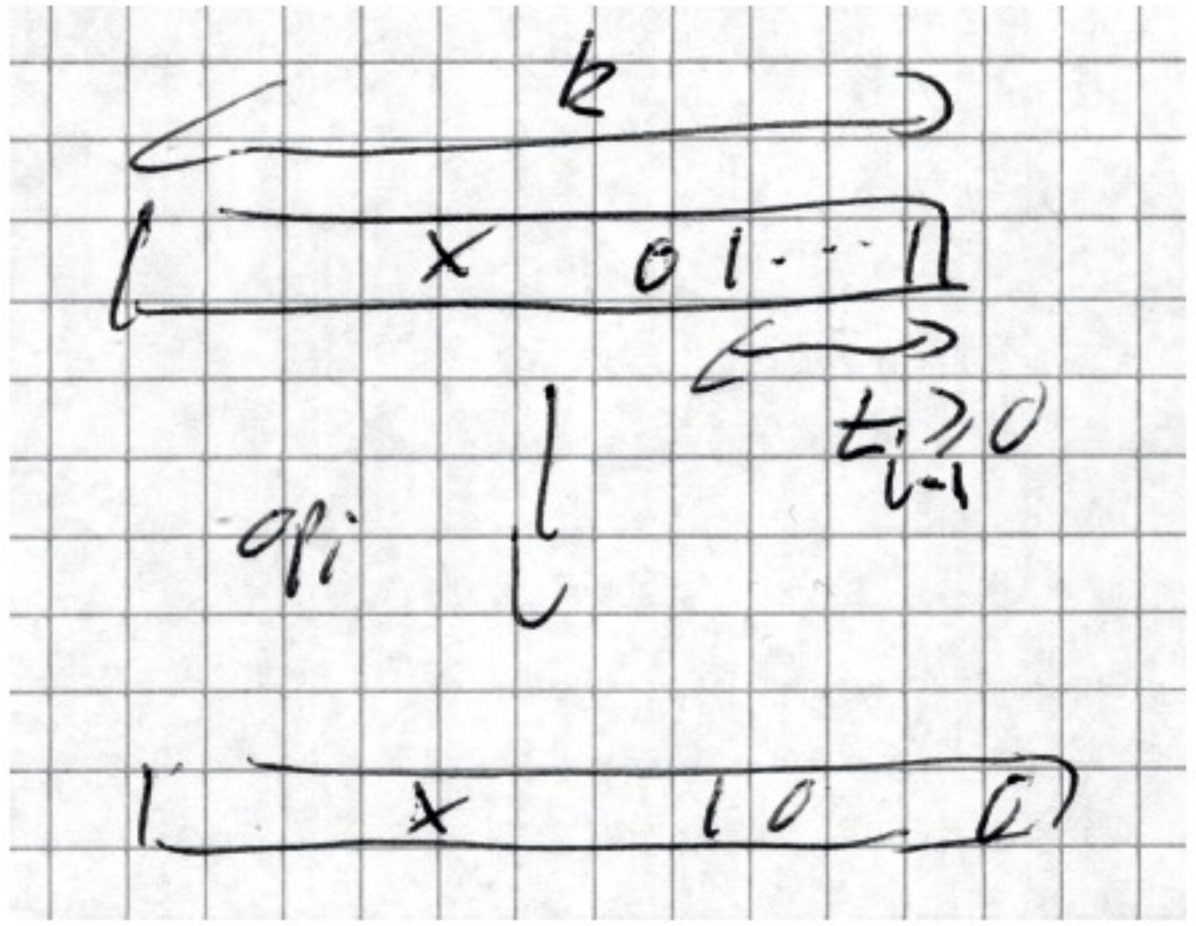

Figure 1: t_{i-1} trailing bits before operation op_i

$$\boxed{c_i \leq t_{i-1} + 1}$$

example: binary counter

- binary counter with k bits
- op_i: increments counter by 1
- c_i: number of bits changed
- t_i: number of trailing ones after operation op_i

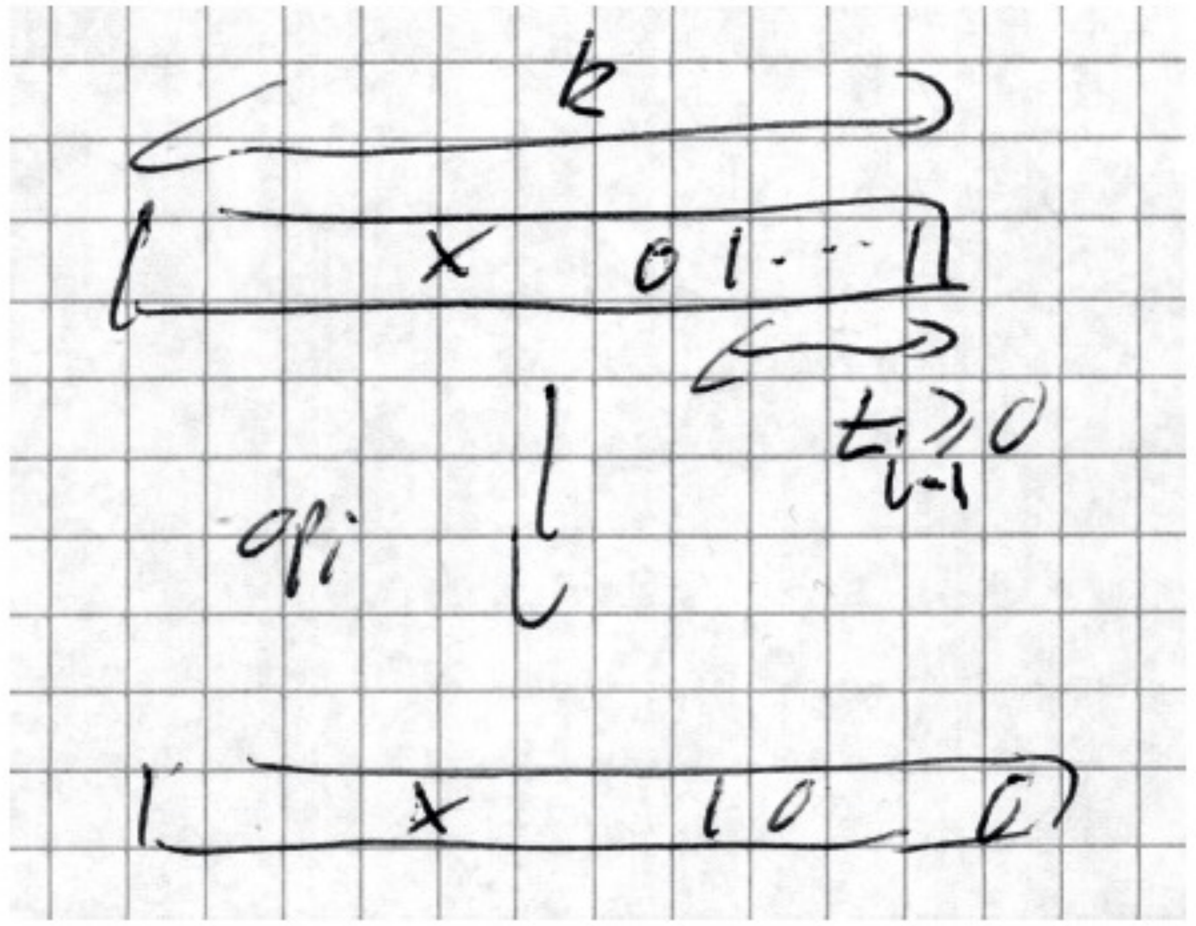

Figure 1: t_{i-1} trailing bits before operation op_i

$$\boxed{c_i \leq t_{i-1} + 1}$$

- Φ_i: number of ones in counter
- $\Phi_i = 0$: all k bits in counter were changed

$$t_{i-1} = \Phi_{i-1} = k\ ,\ \Phi_i < \Phi_{i-1} - t_{i-1} + 1$$

- $\Phi_i > 0$: trailing t_{i-1} ones disapper, 1 new one.

$$\Phi_i = \Phi_{i-1} - t_{i-1} + 1$$

example: binary counter

- binary counter with k bits
- op_i: increments counter by 1
- c_i: number of bits changed
- t_i: number of trailing ones after operation op_i

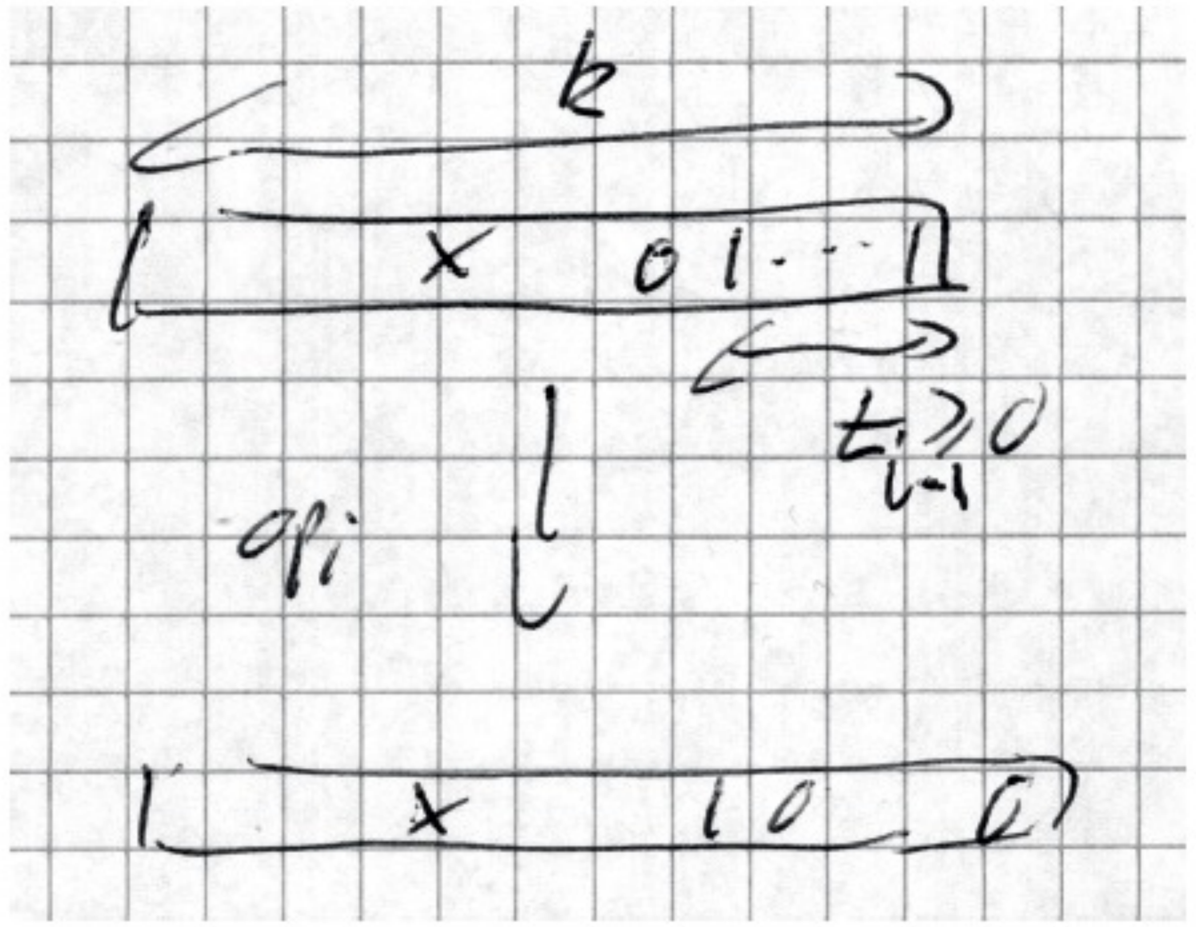

Figure 1: t_{i-1} trailing bits before operation op_i

$$\boxed{c_i \leq t_{i-1} + 1}$$

- Φ_i: number of ones in counter
- $\Phi_i = 0$: all k bits in counter were changed

$$t_{i-1} = \Phi_{i-1} = k\ ,\ \Phi_i < \Phi_{i-1} - t_{i-1} + 1$$

- $\Phi_i > 0$: trailing t_{i-1} ones disapper, 1 new one.

$$\Phi_i = \Phi_{i-1} - t_{i-1} + 1$$

$$\begin{aligned}
\Phi_i - \Phi_{i-1} &\leq -t_{i-1} + 1 \\
&= 1 - t_{i-1} \\
\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\
&\leq 1 + t_{i-1} + 1 - t_{i-1} \\
&= 2
\end{aligned}$$

example: binary counter

- binary counter with k bits
- op_i: increments counter by 1
- c_i: number of bits changed
- t_i: number of trailing ones after operation op_i

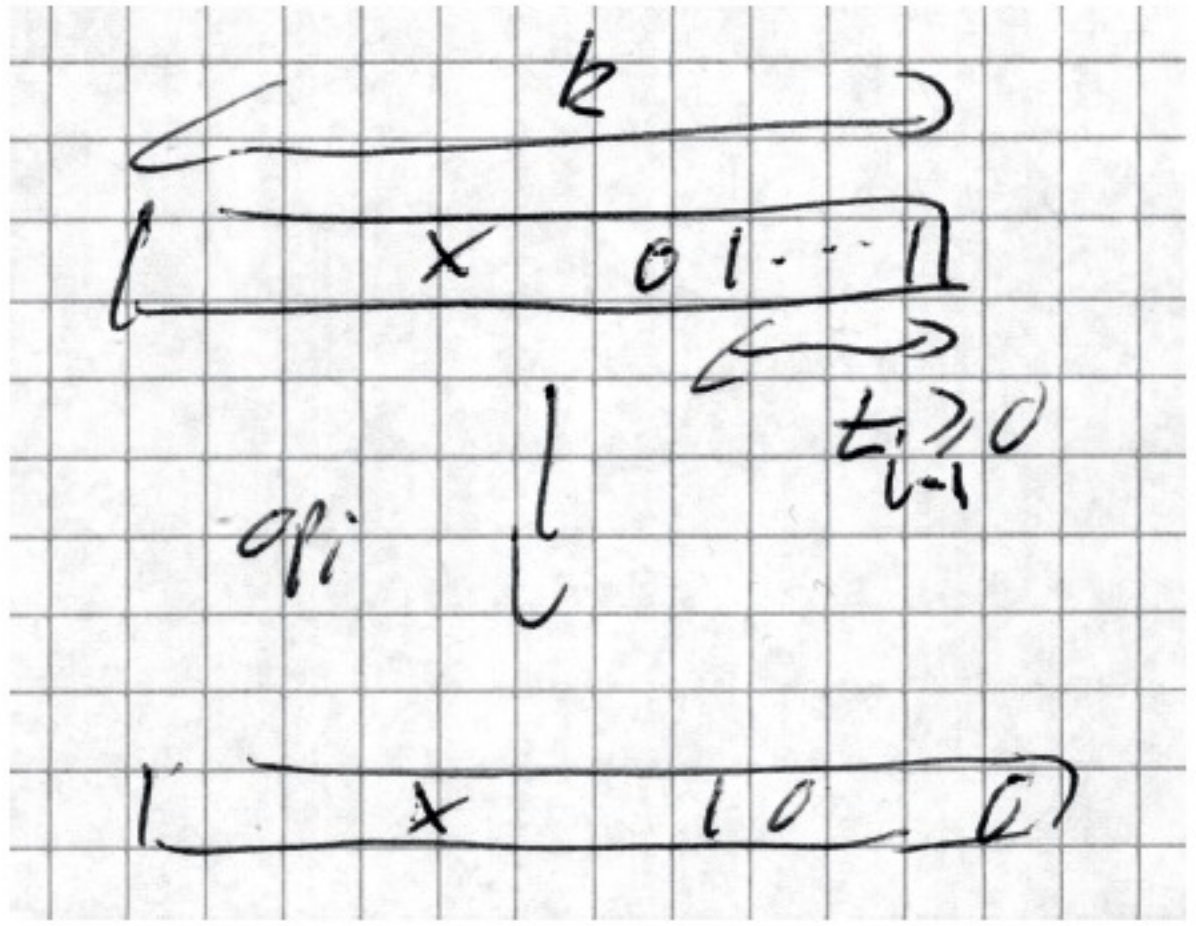

Figure 1: t_{i-1} trailing bits before operation op_i

$$\boxed{c_i \leq t_{i-1} + 1}$$

- Φ_i: number of ones in counter
- $\Phi_i = 0$: all k bits in counter were changed

$$t_{i-1} = \Phi_{i-1} = k \ , \ \Phi_i < \Phi_{i-1} - t_{i-1} + 1$$

- $\Phi_i > 0$: trailing t_{i-1} ones disapper, 1 new one.

$$\Phi_i = \Phi_{i-1} - t_{i-1} + 1$$

$$\begin{aligned}
\Phi_i - \Phi_{i-1} &\leq -t_{i-1} + 1 \\
&= 1 - t_{i-1} \\
\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\
&\leq 1 + t_{i-1} + 1 - t_{i-1} \\
&= 2
\end{aligned}$$

$$\boxed{\begin{aligned}\sum_{i=1}^{n} c_i &\leq \Sigma_{i=1}^{n} 2 + \Phi_0 - \Phi_n \\ &\leq 2n + k\end{aligned}}$$

as $(\Phi_0 - \Phi_n)$ is at most k