Potential Method:

→ We start with an initial data structure D_0 on which we perform n operations.

For each $i = 1, 2, \ldots, n$ let

c_i = the actual cost of ith operation.

D_i = Data structure that results after applying ith operation to D_{i-1}.

ϕ = Potential function that maps each data structure D_i to a real number. $\phi(D_i)$

the potential associated with data structure D_i.

$\hat{c}_i$ = Amortized cost of the ith operation w.r.t function ϕ.

$$\hat{c}_i = \underbrace{c_i}_{\text{actual cost}} + \underbrace{(\phi(D_i) - \phi(D_{i-1}))}_{\text{increase in potential due to ith operation}}$$

The total amortized cost of n op^n is

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} (c_i + \phi(D_i) - \phi(D_{i-1}))$$

$$= \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} (\phi(D_i) - \phi(D_{i-1}))$$

$$= \sum_{i=1}^{n} c_i + \phi(D_n) - \phi(D_0)$$

If we can ensure that $\phi(D_i) \geq \phi(D_0)$

$$\boxed{\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i}$$

The total amortized cost is an upperbound on the total actual cost

However $\phi(D_n) \geq \phi(D_0)$, should hold for all possible n since, in practice, we do not always know $\underline{n}$ in advance.

If $\phi(D_i) - \phi(D_{i-1}) > 0$, then the amortized cost $\hat{c}_i$ represents

- an overcharge to the ith operation
- the potential of the data structure.

If $\phi(D_i) - \phi(D_{i-1}) < 0$, then the amortized cost of $\hat{c}_i$ represents to the ith op^n

- an undercharge
- the actual cost of the op^n is paid by the decrease in potential.