

- Dynamic array

\square
 capacity = 1 = c
 length = 0 = l

add(1) \square c = l = 1

add(2) $\square \square$ c = l = 2

add(3) $\square \square \square$ c = 4
l = 3

add(4) $\square \square \square \square$ c = l = 4

\leftarrow add(5) $\square \square \square \square \square \square \square \square$ c = 8 l = 5

add(6) add(7) add(8)

● light operation

cost 1

● heavy operation

$i = 2^k + 1$

cost $2^k + 1$

before: c = 2^k l = 2^k

after: c = 2^{k+1} l = $2^k + 1$

- aggregate method.

$$\text{amortized cost of add} = \frac{\sum_{i=1}^n \text{cost for } i\text{-th add operation}}{n}$$

$$n \times \text{amortized cost} = \underbrace{\sum_{k: 2^k + 1 \leq n} (2^k + 1)}_{\text{total cost for heavy op}} + \underbrace{\sum_{\substack{i \leq n \\ i \neq 2^k + 1}} 1}_{\text{total cost for light op.}}$$

$$\begin{aligned} \text{assume } n = 2^t + 1 &= \sum_{k=0}^t (2^k + 1) + \sum_{\substack{i \leq n \\ i \neq 2^k + 1}} 1 \\ &= \sum_{k=0}^t 2^k + \sum_{\substack{i \leq n \\ i \neq 2^k + 1}} 1 \\ &\quad \underbrace{2^{t+1} - 1 = 2n - 3}_n \quad \underbrace{\phantom{2^{t+1} - 1 = 2n - 3}}_n \\ &= 3n - 3 \end{aligned}$$

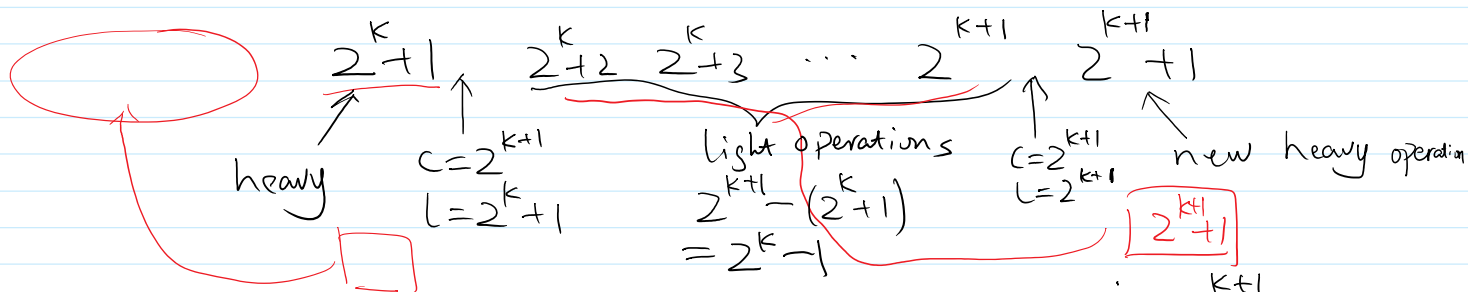
$$n \times \text{amortized cost} = 3n - 3$$

$$\text{amortized cost} = \frac{3n - 3}{n} \leq 3 = O(1)$$

$$\text{amortized cost} = \frac{S^{u-1}}{n} \leq \underline{3} = \underline{O(1)}$$

- accounting (charging) method

- between two heavy operation



idea: to pay for the heavy operation $i = 2^{k+1} + 1$

charge some of that to the light operations in between

$$\text{cost of heavy} \rightarrow \frac{2^{k+1} + 1}{2^k - 1} \approx 2$$

of light op. charging to

charge 2 for every light operation

$$\text{new cost for light operation: } 1 + 2 = 3$$

$$\text{new cost for heavy operation: } 2^{k+1} + 1 - \frac{2(2^k - 1)}{1} = 3$$

\Rightarrow after charging, every operation has cost 3,
amortized cost = $O(1)$

Potential argument.

Φ : function on the state of the system

c : capacity

l : length

$$\Phi(c, l) = \underline{2l} - c$$

light operation $T=1$ $(c, l) \rightarrow (c, l+1)$

$$A = T - (2l - c) + (2(l+1) - c) \quad (\text{potential} + 2)$$

$$= 3$$

$$= 3$$

heavy operation

$$i = 2^k + 1 \quad T = 2^k + 1$$

$$(2^k, 2^k) \rightarrow (2^{k+1}, 2^{k+1})$$

$$A = T - (2 \times 2^k - 2^k) + (2 \times (2^k + 1) - 2^{k+1})$$

$$= 2^k + 1 - 2^k + 2 = 3 \quad (\text{potential } - (2^k - 2))$$