- \((2^{k+1})\)-th add operation
  
  running time \(\geq 2^{k+1}\)  
  
  (allocating an array of size \(2^{k+1}\))
  
  (copy first \(2^k\) elements)
  
  (add the \(2^{k+1}\)th element)

- all other add operation  
  
  running time 1  
  (change length, put the element into an empty slot)

- potential argument

\[
A_i = T_i - \Phi(x_i) + \Phi(x_{i+1})
\]

\[
= T_i - (\Phi(x_i) - \Phi(x_{i+1}))
\]

potential difference

\[
A_1 = T_1 - \Phi(x_1) + \Phi(x_2)
\]

\[
A_2 = T_2 - \Phi(x_2) + \Phi(x_3)
\]

\[
\vdots
\]

\[
A_n = T_n - \Phi(x_n) + \Phi(x_{n+1})
\]

\[
\sum_{i=1}^{n} A_i = \sum_{i=1}^{n} T_i - \Phi(x_i) + \Phi(x_{i+1})
\]

\[
\text{total amortized total runtime}
\]

- design the potential function

- idea: first make sure \(\Phi(x_i) - \Phi(x_{i+1})\) is large

  if \(i = 2^{k+1}\) is a heavy operation. \(T_i = 2^{k+1}\)

\[
\Phi\text{ is a function of } \#\text{ elements and } \text{capacity}
\]
\( \Phi \) is a function of \( \frac{\text{# elements}}{L} \) and \( \frac{\text{capacity}}{C} \)

\[ X_i \xrightarrow{2^k} 2^k \]

Before \( 2^{k+1} \) add operation, data structure has \( 2^k \) elements.

\[ X_{i+1} \xrightarrow{2^{k+1}} 2^{k+1} \]

Difference \( 1 \xrightarrow{2^k} \)

\[ \Rightarrow \text{if capacity } \uparrow \text{ by } 2^k, \text{ potential } \Phi \downarrow \text{ by } 2^{k+1} \]

\[ \Phi = \varnothing - 2 \cdot C \]

- Step 2: \( \Phi \geq 0 \)

Observation: \( L \cdot 2 \geq C \) (except for the initial state)

\[ 1 + 2L \geq C \] (always true)

Can use \( \Phi = 2 + 4L - 2C \) and we know \( \Phi \geq 0 \)

- Compute amortized running time

\( 1 \) heavy operation \( i = 2^{k+1} \), \( T_i = 2^{k+1} \)

\[ \Phi(X_i) = 2 + 4 \cdot (2^k) - 2 \cdot (2^k) \]

\[ \Phi(X_{i+1}) = 2 + 4 \cdot (2^{k+1}) - 2 \cdot (2^{k+1}) \]

\[ \Phi(X_i) - \Phi(X_{i+1}) = 2^{k+1} - 4 \]

\[ A_i = T_i - (\Phi(X_i) - \Phi(X_{i+1})) = 2^{k+1} - (2^{k+1} - 4) = 4 \]

\( 2 \) light operation \( i = 2^{k+1} \), \( T_i = 1 \)

\[ L \xrightarrow{\varnothing} C \]

\[ X_i \xrightarrow{i-1} C \xrightarrow{2+4(i-1) - 2C} X_{i+1} \xrightarrow{i} C \xrightarrow{2+4i - 2C} \]

\[ \Phi(X_i) - \Phi(X_{i+1}) = -4 \]

\[ A_i = T_i - (\Phi(X_i) - \Phi(X_{i+1})) = 1 - (-4) = 5 \]
amortized cost = \( S = O(1) \)

- Union-Find
  - recall: Kruskal
    - for each edge: want to know whether adding the edge creates a cycle.
  - Sets \( \leftrightarrow \) connected components

```
    1
   / \ 5
  2   3
   \ / \
    4
```

add an edge \( \leftrightarrow \) union on the two connected components

edge creates a cycle \( \leftrightarrow \) \( \text{Find}(u) = \text{Find}(v) \)

(\( u, v \)) \( u, v \) in the same set.

- edge 1: \( \text{Find}(1) = 1 \) \( \text{Find}(2) = 2 \)
  \( \{1\} \{2\} \{3\} \{4\} \{5\} \)
  union(1,2)
  \( \{1, 2\} \{3\} \{4\} \{5\} \)

- edge 2: \( \text{Find}(3) = 3 \) \( \text{Find}(4) = 4 \)
  union(3,4)
  \( \{1, 2\} \{3, 4\} \{5\} \)

- edge 3: \( (1, 3) \) \( \text{Find}(1) = 1 \) \( \text{Find}(3) = 3 \)
  union(1,3)
  \( \{1, 2, 3, 4\} \{5\} \)

- edge 4: \( (2, 4) \) \( \text{Find}(2) = 1 \) \( \text{Find}(4) = 1 \)
  do not add the edge