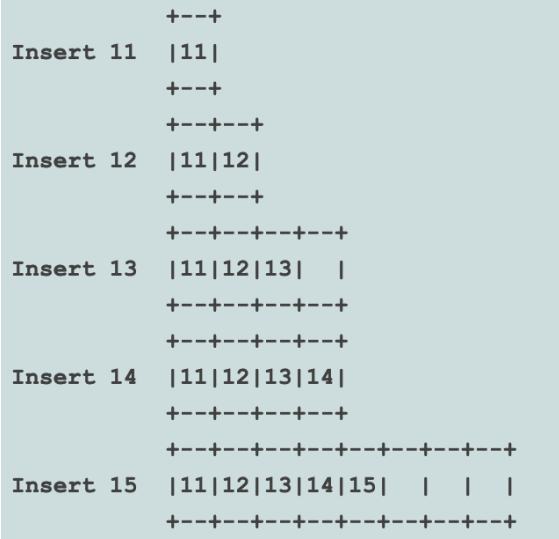
Amortization Analysis (AA)

What is Amortization Analysis?
It is the _____-case analysis of a sequence of operations to obtain _____ bound on the _____ cost per operation in the sequence.

How is it different from the average case analysis? Which one is input-dependent?

Example, Insert to an array with doubling size when array is full. Initially array size is 0 and for each step we insert one item in the array.



What is the time complexity of insert?
Copying over happens after step 2 (copy 1), step 3 (copy 2), step 5 (copy 4), etc…

There are 3 methods in Amortization Analysis.

1. _____ method:
c_i :actual cost of insert at step i
d_i :cost of copying over at step i
$c_i = \sum$ (Express c_i in terms of d_i).

Solve the equation directly.

What is the amortized cost of n inserts?

2. _____ method:
find a _____ charged to each individual operation such that
the _____ for seq of n operation ≤ sum of n _____.

c_i: actual cost of insert at step i
c'_i:
b_i:
At any given step, _____ has to stay non-zero!

$$\Sigma \quad \leq \Sigma$$

Total _____ provides the upper bound for the _____.
How much should be charge for each insert?

What happens when we charge $1?

i	1	2	3	4	5	6	7	8	9	10
s_i	1	2	4	4	8	8	8	8	16	16
c_i	1	2	3	1	5	1	1	1	9	1

What happens when we charge $2?

What happens when we charge $3?

Why $3? How is each $1 spent?

Using accounting method, what is the amortized cost of n inserts?

3. _____ method: Defines potential function for the _____.
Let c_i be the _____
D_i be the _____ of the _____ after i^{th} operation
ϕ_i (D_i): Potential function that maps D_i to a real number,
which represents potential of D_i after i^{th} operation.

- Typically ϕ_0 (D_0) = _____.

Let a_i be the _____

- $a_i = c_i +$
  - What does the second term represent?

- $\sum a_i =$

- What do we need ensure about ϕ_n (D_n) to provide the upper bound for $\sum c_i$?

Define a potential function in a clever way is the key…

Resizing array example: ϕ_i (D_i) = 2N – M

- N:
- M:

Two cases

- Case 1: No need to resize.
  - What is c_i in this case?
  - After ith insert, N → and M →
  - a_i =

- Case 2: Need to resize. When does this happen?
  - What is c_i in this case?
  - After ith insert, N → and M →
  - a_i =

Using potential method, what is the amortized cost for n inserts?

4. Example 1: Stack with multipop

push(x) //Push x onto the stack
x = pop() //Pop x from the top of the stack
multipop(k) //Pop the k topmost elements off of the stack.
//If the stack contains fewer than k items, pop the entire stack.

What is the amortized cost for each operation?

Use accounting method:

Use potential method:

5. **Example 2: Even more dynamic array**

A data structure that supports deletes can both grow and shrink in size. It would be nice if the size that it occupies is not too much bigger than necessary. This is where a list that shrinks in size is useful.

Let c be the capacity of the array.
Let n be the number of actual list elements.

- initialize(): create an empty list with capacity 2. (c = 2 and n = 0)
- append(): if c = n then grow(). Now insert the element into the table.
- pop(): if n = c /4 and c ≥ 4 then shrink(). Now erase the element from the table

Where
- grow() increases the capacity of the array from c to 2c ,
- shrink() decreases the capacity of the array from c to c /2.

Cost model:
- Writing a value to any location in an array costs 1
- Moving a value from one array to another costs 1
- Erasing a value from an array costs 1
- All other operations are free

What is the amortized cost of `append`, `pop`, and `initialize`?
Use potential method.

Note: c /2 is the "midpoint" in some sense of the capacity,
since we will always have n = c /2 immediately following a grow or a shrink.
So, it would be nice if the potential function still had the property that it was equal to zero when n = c /2.