

All-operations Constant - time

implementation of

ADT Array

Operations are :

`init()` - initializes a new array. Old one is destroyed. New one has no elements in it.

`set(int i, float val)` - next "`get(i)`" will return `val`

`float get(int i)`

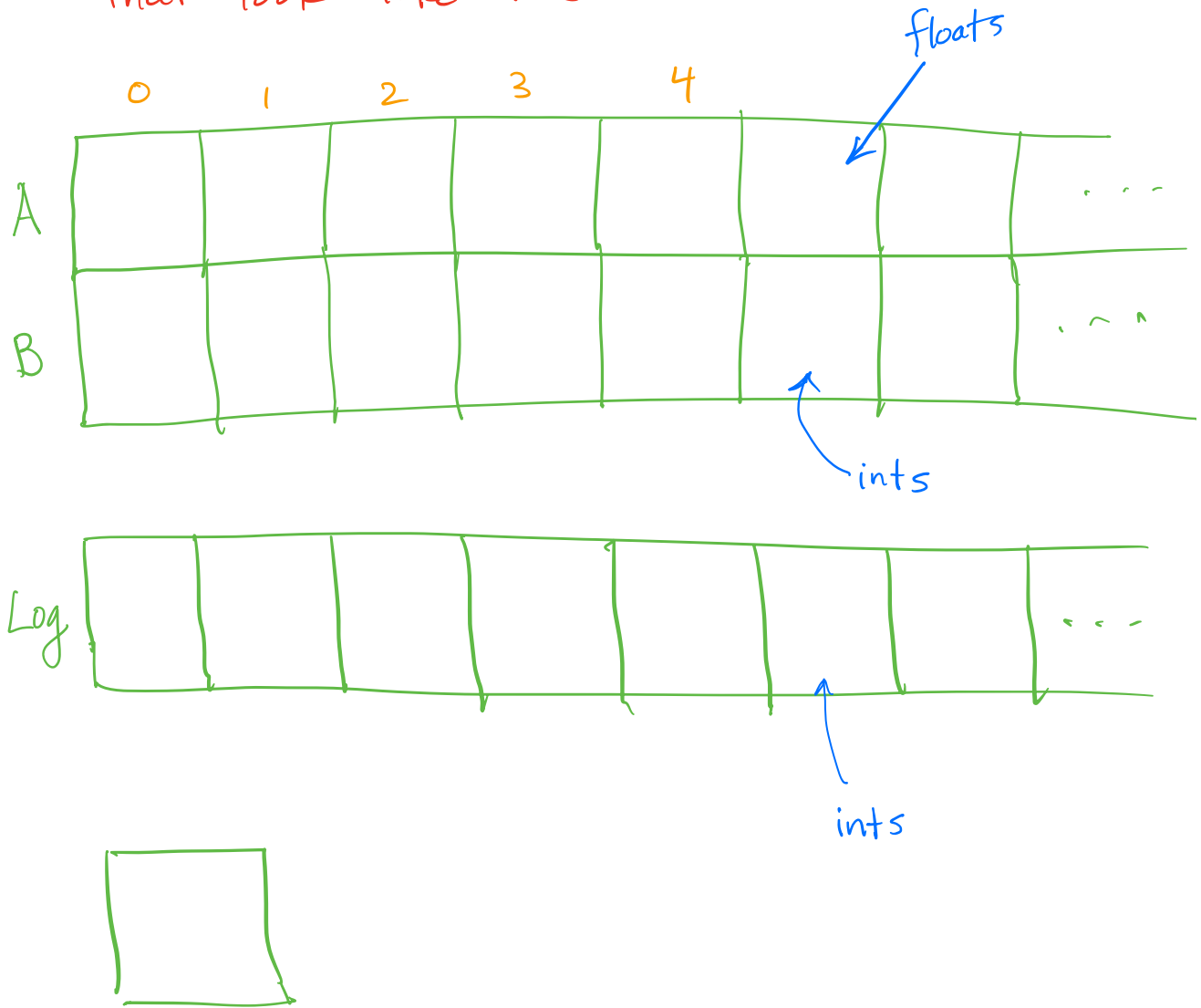
- if there has been no "`set(i, val)`" since last "`init()`", returns Sentinel value `UNDEF`
- else returns `val` of most recent "`set(i, val)`"

Assume : `init` can allocate an arbitrarily large amount of space (hint: for multiple C++ arrays) in constant time

- such space will be full of garbage

- We will assume the space allocated is sufficient.

You can do it with Data Structures
that look like this :



↑ an integer m which is an index into Log