discordDucks (Marcus Wu + Nat, Gabriel Thompson + Iggy, Kartik Vanjani + Krrish) (Green)
APCS1 pd8
HW46: Wrap the Wrapper
2021-12-09t
time spent: 0.5hrs

Reviewed By Team MumbleCore

wrapper class Foo

boolean add(int) → adds the int to the end, but only if maintains ascending order. Returns true

boolean add(int, int) → adds the second at the first index, but only if maintains ascending order. Returns true

void remove(int) → removes the item at the inputted index, and shifts the array corresponding. Only does so if it maintains ascending order

int set(int, int) → sets the value at the index of the first int to the second int. Returns the old value

int size() → returns the size of the array

void expand() → doubles the size of the array, filling the second half with zeros

String toString() → returns a String representation of the instance

void expand() → doubles the size of the array, filling the second half with zeros

int get(int) → returns the value at the index

cool

I agree

is SuperArray?!!

class SuperArray

NOTE/Concerns: We weren't sure if we were supposed to include instance variables and whatnot. We did not, and we just wrote the functions from SuperArray, but with checks to make sure that it didn't violate ascending order

Same

Team: Silly serpents

Members:

Jason Zhou + Tiggy,

Vansh Saboo + Tiffany

discordDucks (Marcus Wu + Nat, Gabriel Thompson + Iggy, Kartik Vanjani + Krrish)
APCS1 pd8
HW46: Wrap the Wrapper
2021-12-09t
time spent: 0.5hrs

where one reference vars?

wrapper class Foo

- boolean add(int) → adds the int to the end, but only if maintains ascending order. Returns true
- void expand() → doubles the size of the array, filling the second half with zeros
- boolean add(int, int) → adds the second at the first index, but only if maintains ascending order. Returns true
- String toString() → returns a String representation of the instance
- void remove(int) → removes the item at the inputted index, and shifts the array corresponding. Only does so if it maintains ascending order
- void expand() → doubles the size of the array, filling the second half with zeros
- int set(int, int) → sets the value at the index of the first int to the second int. Returns the old value
- int get(int) → returns the value at the index
- int size() → returns the size of the array

return false if not ascending order

↓

class SuperArray

NOTE/Concerns: We weren't sure if we were supposed to include instance variables and whatnot. We did not, and we just wrote the functions from SuperArray, but with checks to make sure that it didn't violate ascending order