Tradeoffs

What makes the circular queue implementation different from the linked list implementation? In what ways is one more desirable than the other?

10

10

Example

What is the worst case number of simple operations for this piece of code?
Let `A` have n entries.

Linear search

```
int linearSearch(int[] A, int target){
  for(int i = 0; i < A.length; i++){
    if(A[i] == target)
      return i;
  }
  return -1;
}
```

21

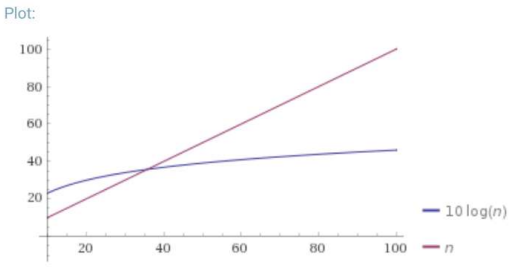
21

Why is that the definition?

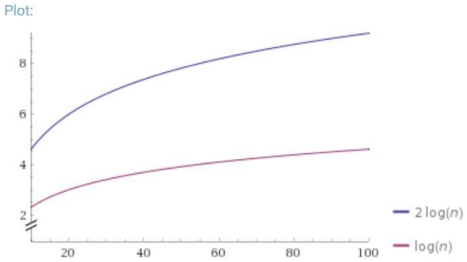
Big-O

$f(n)$ is $O(g(n))$ if there exist positive constants c, n_0 such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$

Why n_0?



Why c?



25

25

O, Omega, Theta [oh my?]

Big-O is an **upper bound**

- My code uses at most this many resources (e.g. runs in at most this much time)

Big-Omega is a lower bound

Big-Omega

$f(n)$ is $\Omega(g(n))$ if there exist positive constants c, n_0 such that for all $n \geq n_0$, $f(n) \geq c \cdot g(n)$

Big Theta is "equal to"

Big-Theta

$f(n)$ is $\Theta(g(n))$ if $f(n)$ is $O(g(n))$ and $f(n)$ is $\Omega(g(n))$.

28

28