

Algo Hw3: Question 2

2-1) Asymptotic Time:  $T(n) = 4T(n/2) + n$

$$T(n) = 4T(n/2) + 2$$

$$\rightarrow T(n) = 4(4T(n/4)) + (n/2) + n = 16T(n/4) + 4n/2 + n$$

$$\rightarrow T(n) = 16(4T(n/8) + n/4) + 4n/2 + n = 64T(n/8) + 16n/4 + 4n/2 + n$$

$$\dots$$

$$\rightarrow T(n) = 4^k T(n/2^k) + kn$$

stop expanding when  $n/2^k=1$ , which gives  $k=\log_2 n$ , so subbing

$$K = T(n) = 4^{\log_2 n} T(1) + (\log_2 n) n$$

$$\hookrightarrow T(n) = O(n^2) \text{ since } 4^{\log_2 n} = n^{\log_2 4} = n^2 \text{ as } T(1) \text{ const}$$

Asymptotic time complexity =  $O(n^2)$

2-2) T/F? proof/counter example

a) Is  $2^{n+1} = O(2^n)$

check:  $2^{n+1} = O(2^n)$

This means there is constant  $c > 0$

$\exists n_0$  such that:

$$\rightarrow 2^{n+1} \leq c \cdot 2^n \text{ for all } n \geq n_0$$

$$\rightarrow 2^{n+1} = 2 \cdot 2^n$$

div by 2  $\rightarrow 2 \leq c$  TRUE

→ True for any constant  $c \geq 2$  so there is a constant that inequality holds

Q 2-3) Arrays list by growth rate

- $f_1 = 10n$
  - $f_2 = n^n$
  - $f_3 = \log_2 n$
  - $f_4 = 2^{\log_2 n}$
  - $f_5 = \sqrt{2n}$
  - $f_6 = n^{2.5}$
- (1) Log growth rates  
Slowest
- (2) Sub polynomial
- (3) Polynomial
- (4) Exponential
- (5)  $10^n$  faster than  $\sqrt{2n}$
- (6)  $n^n$  grows fastest

Final List

$\log_2 n$

$\sqrt{2n}$

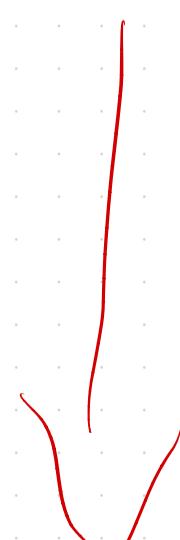
$n^{2.5}$

$\sqrt{2^n}$

$10^n$

$n^n$

Slowest



Fastest

b.)  $2^{2n} = O(2^n)$

this means that there is constant  $C > 0$  and no such that:  $2^{2n} \leq C \cdot 2^n$  for all  $n \geq n_0 =$

$$(2^n)^2 \leq n^2$$

div by 2?  $2^n < C$ , since  $2^n$  grows exponentially, inequality cannot be satisfied for all large  $n$  with a constant  $C$ . No constant can be bound as  $2^n$  because  $n \rightarrow \infty$ . Statement false

Counter Example:  $n = 10$

$$2^{2(10)} = 2^{20} = 1048576$$

$$C \cdot 2^{10} = C \cdot 1024$$

FALSE