

## Assignment Project Exam Help

https://pm/scoder.com

# Add We Chat powcoder

University of New South Wales

3. RECURRENCES - part A

• "Big Oh" notation: f(n) = O(g(n)) is an abbreviation for:

Assignment Project Exam Help  $0 \le f(n) \le c g(n)$  for all  $n \ge n_0$ ".

https://powcoder.com

• "Big Oh" notation: f(n) = O(g(n)) is an abbreviation for:

# Assignment Project Exam Help $0 \le f(n) \le c g(n)$ for all $n \ge n_0$ ".

• In that the say the Owico continue of the first of the say the owice of the own of the say of the

• "Big Oh" notation: f(n) = O(g(n)) is an abbreviation for:

# Assignment Project Exam Help $0 \le f(n) \le c g(n)$ for all $n \ge n_0$ ".

- In the tap's say the Owico of the order of the own of
- $f(n) \land O(g(n))$  means that f(n) does not grow substantially faster than g(n) because a multiple of g(n) eventually dominates f(n).

• "Big Oh" notation: f(n) = O(g(n)) is an abbreviation for:

# Assignment Project Exam Help $0 \le f(n) \le c g(n)$ for all $n \ge n_0$ ".

- In that the say the Owi Co of the order of the own of
- $f(n) \land O(g(n))$  means that f(n) does not grow substantially faster than g(n) because a multiple of g(n) eventually dominates f(n).
- Clearly, multiplying constants c of interest will be larger than 1, thus "enlarging" g(n).

• "Omega" notation:  $f(n) = \Omega(g(n))$  is an abbreviation for:

Assignment Project Exam Help

https://powcoder.com

• "Omega" notation:  $f(n) = \Omega(g(n))$  is an abbreviation for:

# Assignment Project Exam Help

• In this case we say that g(n) is an asymptotic lower bound for f(n) ttps://powcoder.com

• "Omega" notation:  $f(n) = \Omega(g(n))$  is an abbreviation for:

## Assignment Project Exam Help

- In this case we say that g(n) is an asymptotic lower bound for f(n) ttps://powcoder.com
- $f(n) = \Omega(g(n))$  essentially says that f(n) grows at least as fast as g(n), because f(n) eventually dominates a multiple of g(n).

  Add WeChat powcoder

• "Omega" notation:  $f(n) = \Omega(g(n))$  is an abbreviation for:

# Assignine Project Exam Help

- In this case we say that g(n) is an asymptotic lower bound for f(n)https://powcoder.com
- $f(n) = \Omega(g(n))$  essentially says that f(n) grows at least as fast as g(n), because f(n) eventually dominates a multiple of g(n).

  Add WeChat powcoder

  • Since  $c g(n) \le f(n)$  if and only if  $g(n) \le \frac{1}{c} f(n)$ , we have
- $f(n) = \Omega(g(n))$  if and only if g(n) = O(f(n)).

• "Omega" notation:  $f(n) = \Omega(g(n))$  is an abbreviation for:

# Assigning Project Exam Help

- In this case we say that g(n) is an asymptotic lower bound for f(n)https://powcoder.com
- $f(n) = \Omega(g(n))$  essentially says that f(n) grows at least as fast as g(n), because f(n) eventually dominates a multiple of g(n).

  Add WeChat powcoder

  • Since  $c g(n) \le f(n)$  if and only if  $g(n) \le \frac{1}{c} f(n)$ , we have
- $f(n) = \Omega(g(n))$  if and only if g(n) = O(f(n)).
- "Theta" notation:  $f(n) = \Theta(g(n))$  if and only if f(n) = O(g(n))and  $f(n) = \Omega(g(n))$ ; thus, f(n) and g(n) have the same asymptotic growth rate.

• Recurrences are important to us because they arise in estimations

Assignment Project Exam Help

https://powcoder.com

• Recurrences are important to us because they arise in estimations of time complexity of divide-and-conquer algorithms.

Assignment Project Exam Help

MERGE SORT (A, p, r)

\*sorting A[p..r]\*

- thereps://powcoder.com
- Merge Sort  $(A, a \perp 1, r)$
- Merge-Sort(A, q+1, r)
- Add WeChat powcoder

• Recurrences are important to us because they arise in estimations of time complexity of divide-and-conquer algorithms.

# Assignment Project Exam Help Merge Sort(A, p, r) Assignment Project Exam Help

- thereps://powcoder.com
- Manga Cont (A, a + 1, b)
- Merge-Sort(A, q+1, r)
  - Add We Chat powcoder
  - Since  $\operatorname{Merge}(A, p, q, r)$  runs in linear time, the runtime T(n) of  $\operatorname{Merge-Sort}(A, p, r)$  satisfies

$$T(n) = 2T\left(\frac{n}{2}\right) + c n$$

• Let  $a \ge 1$  be an integer and b > 1 a real number;

## Assignment Project Exam Help

https://powcoder.com

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignment Project Exam Help

https://powcoder.com

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignment Project Examilielp

https://powcoder.com

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignmental of sprojectary per period property of splitting up/combining the solutions for size n/b into a solution for size n is if f(n),

https://powcoder.com

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assign that of the continuity per property per property problems of size n/b into a solution for size n is if f(n),

 then the time complexity of such algorithm satisfies https://powcoder.com

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignment of the officer of course problems of size p obtains the solutions for size p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p is if p into a solution for size p in the solution p into a solution for size p is in the solution p into a solution for size p in the solution p is a solution for size p in the solution p into a solution for size p in the solution p is a solution for size p in the solution p in the solution p in the solution p is a solution for size p in the solution p in the solution p in the solution p is a solution p in the solution p in t

• then the time complexity of such algorithm satisfies

- Let a > 1 be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

SSign appropriate of the control of the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions for size per overnead cost of splitting up/combining the solutions of size per overnead cost of splitting up/combining the solutions of size per overnead cost of splitting up/combining the solutions of size per overnead cost of splitting up/combining n/b into a solution for size n is if f(n),

• then the time complexity of such algorithm satisfies

https://powcoder.com

· Note Addul We Chat powcoder

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignment of the continuity of the solutions for size n/b into a solution for size n/b

• then the time complexity of such algorithm satisfies

• Note Addul We Ciphat powcoder

$$T(n) = a T\left(\left\lceil \frac{n}{b}\right\rceil\right) + f(n)$$

- Let  $a \ge 1$  be an integer and b > 1 a real number;
- Assume that a divide-and-conquer algorithm:

Assignment of the continuity period period

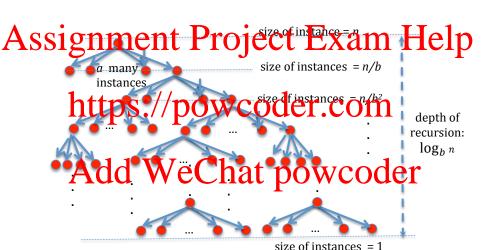
• then the time complexity of such algorithm satisfies

• Note Addul We Chat powcoder

$$T(n) = a T\left(\left\lceil \frac{n}{b}\right\rceil\right) + f(n)$$

but it can be shown that ignoring the integer parts and additive constants is OK when it comes to obtaining the asymptotics.

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$



• Some recurrences can be solved explicitly, but this tends to be tricky.

### Assignment Project Exam Help

https://powcoder.com

 Some recurrences can be solved explicitly, but this tends to be tricky.

Assignmental Project a Exhamolie p

https://powcoder.com

 Some recurrences can be solved explicitly, but this tends to be tricky.

# Assignmentat Project a Exhamolie potential exact solution of a recurrence

• We hittps://powcoder.com

 Some recurrences can be solved explicitly, but this tends to be tricky.

# Assignmentateffeetect acreamentely the exact solution of a recurrence

• We ply ped to find owcoder.com the growth rate of the solution i.e., its asymptotic behaviour;

• Some recurrences can be solved explicitly, but this tends to be tricky.

# Assignmentateffcetect acoxiamone por the exact solution of a recurrence

- We planted to find owcoder. Combine of the solution i.e., its asymptotic behaviour;

  - the (approximate) sizes of the constants involved (more about that later)

• Some recurrences can be solved explicitly, but this tends to be tricky.

#### Assignmental Project & Exame Help the exact solution of a recurrence

- We ply ped to find owcoder.com
  the growth rate of the solution i.e., its asymptotic behaviour;

  - the (approximate) sizes of the constants involved (more about that later)

Add WeChat powcoder

This is what the Master Theorem provides (when it is

applicable).

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Assignment Project Exam Help

https://powcoder.com

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Assignment Project Exam Help

https://powcoder.com

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Assignment ectrocjecta Exam Help

https://powcoder.com

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Associated the lectron of the lectro

https://powcoder.com

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Asson the anon-decreasing unction; ecta Exam Help

1 If  $f(n) = O(n^{\log_b a - \varepsilon})$  for some  $\varepsilon > 0$ , https://powcoder.com

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

Assignment he lectrodie et a Exam Help

1 If  $f(n) = O(n^{\log_b a - \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ; https://powcoder.com

Let:

- a > 1 be an integer and and b > 1 a real;
- Assing the anon-decreasing function ect. Exam Help Then:

  - 1 If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
    2 If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $O(n) = O(n^{\log_b a})$ ;

Let:

• a > 1 be an integer and and b > 1 a real;

Assing the anon-decreasing function ect. Exam Help Then:

- 1 If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
  2 If  $f(n) = O(n^{\log_b a \varepsilon})$ , the power constant.

Let:

• a > 1 be an integer and and b > 1 a real;

\$\$\frac{f(n)}{2}\text{polybeanon-decreasin Punction;} \text{ect}\_a \text{Exam Help}

Then:

- 1 If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
  2 If  $f(n) = O(n^{\log_b a + \varepsilon})$ , the property of the constant  $\varepsilon > 0$ , then  $T(n) = O(n^{\log_b a})$ ;
  3 If  $f(n) = O(n^{\log_b a + \varepsilon})$  for some  $\varepsilon > 0$ ,

Let:

• a > 1 be an integer and and b > 1 a real;

\$\$\frac{f(n)}{2}\text{polybeanon-decreasin Punction;} \text{ect}\_a \text{Exam Help}

Then:

- If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
- 2 If  $f(n) = \Omega(n^{\log_b a + \varepsilon})$  for some  $\varepsilon > 0$ , and for some c < 1 and some  $n_0$ ,

holds fold d d d d o WeChat powcoder

Let:

• a > 1 be an integer and and b > 1 a real;

\$\$\frac{f(n)}{2}\text{polybeanon-decreasin Punction;} \text{ect}\_a \text{Exam Help}

Then:

- If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
- If  $f(\mathbf{n} + \mathbf{n}) = \Omega(\mathbf{n}^{\log_b a + \varepsilon})$  for some  $\varepsilon > 0$ , and for some c < 1 and some  $n_0$ ,

 $\text{holds fold } n_0, \text{ then } Ch_{(a,t)}^{a f(n/b) \le c f(n)}$ 

Let:

•  $a \ge 1$  be an integer and and b > 1 a real;

## Assignment Project Exam Help

Then:

- If  $f(n) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
- o If f(https://poweoder.com
- 3 If  $f(n) = \Omega(n^{\log_b a + \varepsilon})$  for some  $\varepsilon > 0$ , and for some c < 1 and some  $n_0$ ,

# $\text{holds} \text{fold} \underset{n_0, \text{then}}{\text{Mod}} \text{d} \underset{n_0, \text{then}}{\text{Mod}} \text{d} \underset{n_0, \text{then}}{\text{Chat}}); \text{powcoder}$

4 If none of these conditions hold, the Master Theorem is NOT applicable.

Let:

• a > 1 be an integer and and b > 1 a real;

## \$50 be a non-decreasin Punction; ecta Exam Help

Then:

- If  $f(\underline{n}) = O(n^{\log_b a \varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b a})$ ;
- If  $f(\mathbf{n} + \mathbf{n}) = \Omega(\mathbf{n}^{\log_b a + \varepsilon})$  for some  $\varepsilon > 0$ , and for some c < 1 and some  $n_0$ ,

# $\text{holds fold } \underset{n_0,\text{then } \textbf{Chart}}{\text{holds fold }} \text{holds fold } \underset{n_0,\text{then } \textbf{Chart}}{\text{holds fold }} \text{powcoder}$

If none of these conditions hold, the Master Theorem is NOT applicable.

(But often the proof of the Master Theorem can be tweaked to obtain the asymptotic of the solution T(n) in such a case when the Master Theorem does not apply; an example is  $T(n) = 2T(n/2) + n \log n$ .

• Note that for any b > 1,

Assignment  $\overset{\log_b n = \log_b 2 \log_2 n}{\text{Project Exam Help}}$ 

https://powcoder.com

• Note that for any b > 1,

Assignment 
$$\underset{c = \log_b 2 > 0}{\operatorname{log}_b n = \log_b 2 \log_2 n}$$
;

 $https://pow_{c} \stackrel{\log_b n = c}{\underset{c}{\log_2 n}}; com$ 

• Note that for any b > 1,

Assignment  $\underset{c = \log_b 2 > 0}{\log_b n = \log_b 2 \log_2 n}$ ;

 $https://pow_{c} \stackrel{\log_b n = c}{\underset{c}{\log_2 n}}; com$ 

• Thus,

• Note that for any b > 1,

Assignment  $\underset{c = \log_b 2 > 0}{\log_b n = \log_b 2 \log_2 n}$ ;

 $https://pow_{c} \stackrel{\log_b n = c}{\underset{c}{\log_2 n}}; com$ 

• Thus,

# and Add We $\operatorname{Chat}_{\log_2 n}^{\log_2 n} \operatorname{powcoder}_{\log_2 n}$

• So whenever we have  $f = \Theta(g(n) \log n)$  we do not have to specify what base the log is - all bases produce equivalent asymptotic estimates (but we do have to specify b in expressions such as  $n^{\log_b a}$ ).

• Let T(n) = 4T(n/2) + n;

## Assignment Project Exam Help

https://powcoder.com

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help

https://powcoder.com

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

https://powcoder.com

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

Conlittps://powcoder.com

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

Conlittps://ipowcoder.eom

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

### Conlittps://powcoder.eom

• Let T(n) = 2T(n/2) + 5n;

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

### Conlittps://powcoder.eom

• Let T(n) = 2T(n/2) + 5n;

then And I We Chat powcoder

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

### Conlittps://ipowcoder.eom

• Let T(n) = 2T(n/2) + 5n;

### then And In We Chat powcoder

thus 
$$f(n) = 5 n = \Theta(n) = \Theta(n^{\log_2 2})$$
.

• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

### Conlittps://ipowcoder.eom

• Let T(n) = 2T(n/2) + 5n;

### then And In We Chat powcoder

thus 
$$f(n) = 5 n = \Theta(n) = \Theta(n^{\log_2 2})$$
.

Thus, condition of case 2 is satisfied;



• Let T(n) = 4T(n/2) + n;

# Assignment Project Exam Help thus $f(n) = n = O(n^{2-\varepsilon})$ for any $\varepsilon < 1$ .

### Conlittps://powcoder.eom

• Let T(n) = 2T(n/2) + 5n;

### then And In We Chat powcoder

thus 
$$f(n) = 5 n = \Theta(n) = \Theta(n^{\log_2 2})$$
.

Thus, condition of case 2 is satisfied; and so,

$$T(n) = \Theta(n^{\log_2 2} \log n) = \Theta(n \log n).$$

• Let T(n) = 3T(n/4) + n;

## Assignment Project Exam Help

https://powcoder.com

```
• Let T(n) = 3T(n/4) + n;
• then n^{\log_b a} = n^{\log_4 3} < n^{0.8};
```

## Assignment Project Exam Help

https://powcoder.com

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ;
- Assignment Project Exam Help

https://powcoder.com

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ;
- Assignment (Project Exam Help

https://powcoder.com

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ :
- Assignment (Project Exam Help

https://powcoder.com

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ; • thus  $f(n) = n - \Omega(n^{28+\epsilon})$  for any  $\epsilon < 0.00$
- Assignment Project Exam Help
  - Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .

    https://powcoder.com
    - Add WeChat powcoder

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ;
- Assignment (Project Exam Help
  - Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .
  - . Let https://pow.coder.com

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ;
- Assignment (Project Exam Help
  - Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .
  - . Let https://pow.coder.com
    - then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ; Assign f(n) = n = 1000 for any exam. Help
  - Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .
  - . Let https://pow.coder.com
    - then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .
    - Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ . Add WeChat powcoder

- Let T(n) = 3T(n/4) + n; • then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ :
- Assignment (Project Exam Help
  - Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .
  - . Let https://pow.coder.com
    - then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .

    - Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ . Award  $f(n) = n \log_2 n \equiv \Omega(n)$ .

- Let T(n) = 3T(n/4) + n;
  - then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ :

## Assignment (Project Exam Help

- Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .
- . Let https://pow.coder.com
  - then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .

  - Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ . Thus is because for every  $\varepsilon > 0$ , and every c > 0, no matter how
  - small,  $\log_2 n < c \cdot n^{\varepsilon}$  for all sufficiently large n.

- Let T(n) = 3T(n/4) + n;
  - then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ ;

## Assignment Project Exam Help

• Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .

### . Let https://pow.coder.com

- then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .
- Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ .
- Mwere f(n) f(n)
- This is because for every  $\varepsilon > 0$ , and every c > 0, no matter how small,  $\log_2 n < c \cdot n^{\varepsilon}$  for all sufficiently large n.
- Homework: Prove this.

- Let T(n) = 3T(n/4) + n;
  - then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ :

## Assignment (Project Exam Help

• Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .

### . Let https://pow.coder.com

- then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .
- Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ . Thus is because for every  $\varepsilon > 0$ , and every c > 0, no matter how
- small,  $\log_2 n < c \cdot n^{\varepsilon}$  for all sufficiently large n.
- Homework: Prove this. Hint: Use de L'Hôpital's Rule to show that  $\log n/n^{\varepsilon} \to 0$ .

- Let T(n) = 3T(n/4) + n;
  - then  $n^{\log_b a} = n^{\log_4 3} < n^{0.8}$ :

## Assignment (Project Exam Help

• Thus, Case 3 applies, and  $T(n) = \Theta(f(n)) = \Theta(n)$ .

### . Let https://pow.coder.com

- then  $n^{\log_b a} = n^{\log_2 2} = n^1 = n$ .
- Thus,  $f(n) \equiv n \log_2 n \equiv \Omega(n)$ . However,  $f(n) \neq n = 0$ . This is because for every  $\varepsilon > 0$ , and every c > 0, no matter how
- small,  $\log_2 n < c \cdot n^{\varepsilon}$  for all sufficiently large n.
- **Homework:** Prove this. Hint: Use de L'Hôpital's Rule to show that  $\log n/n^{\varepsilon} \to 0$ .
- Thus, in this case the Master Theorem does **not** apply!



#### Master Theorem - Proof:

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

## Assignment Project Exam Help

https://powcoder.com

#### Master Theorem - Proof:

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

 $Assignment \stackrel{\text{implies (by applying it to } n/b \text{ in place } p_{\overline{b}}^{n})}{\text{Exam Help}}$ 

https://powcoder.com

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

 $Assignment_{(\frac{1}{b})}^{\text{implies (by applying it to } n/b \text{ in place}} \underbrace{P_{ro}}_{b^2} \underbrace{p_{ro}}_{b^2} \underbrace{Exam \ Help}_{b^2}$ 

and (by applying (1) to  $n/b^2$  in place of n)

https://powcoder.com (3)

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

https://powcoder.com (3)

and so on ...,

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

https://powcoder.com (3)

and so on ..., we get

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

# https://powcoder.com

and so on ..., we get

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

# https://powcoder.com

and so on ..., we get

$$T(n) = a \underbrace{A_{\overline{b}}^{n} (1)}_{(2L)} + a f(\frac{n}{b}) + f(n)$$

$$= a^{2} T(\frac{n}{b^{2}}) + a f(\frac{n}{b}) + f(n)$$

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

# https://powcoder.com

and so on  $\dots$ , we get

$$T(n) = a \underbrace{A \frac{n}{b} d d}_{(2L)} \underbrace{Powcoder}_{(2R)} \underbrace{Powcoder}_{(2R)} \underbrace{Powcoder}_{(3L)} \underbrace{+ n \frac{n}{b} d + n \frac{n}{b}}_{(3R)} \underbrace{+ n \frac{$$

Since

$$T(n) = a T\left(\frac{n}{b}\right) + f(n) \tag{1}$$

and (by applying (1) to  $n/b^2$  in place of n)

# https://powcoder.com

and so on ..., we get

$$T(n) = a \underbrace{A \choose b} \underbrace{n}_{(2L)} \underbrace{n}_{(2R)} \underbrace{powcoder}_{(2R)}$$

$$= a^2 T \left(\frac{n}{b^2}\right) + a f\left(\frac{n}{b}\right) + f(n) = a^2 \left(a T \left(\frac{n}{b^3}\right) + f\left(\frac{n}{b^2}\right)\right) + a f\left(\frac{n}{b}\right) + f(n)$$

$$= a^3 T \left(\frac{n}{b^3}\right) + a^2 f\left(\frac{n}{b^2}\right) + a f\left(\frac{n}{b}\right) + f(n) = \dots$$

Continuing in this way  $\log_b n - 1$  many times we get ...

https://powcoder.com

Continuing in this way  $\log_b n - 1$  many times we get ...

$$\mathbf{h}^{=}_{\mathbf{t}} \mathbf{h}^{\lfloor \log_b n \rfloor} T / \mathbf{h}^{\lfloor \frac{n}{b} \rfloor} + a^{\lfloor \log_b n \rfloor - 1} f (\mathbf{h}^{\lfloor \log_b n \rfloor - 1}$$

Continuing in this way  $\log_b n - 1$  many times we get ...

$$\mathbf{h}^{=}_{\mathbf{t}} a^{\lfloor \log_b n \rfloor} T \left( \frac{n}{b^{\lfloor \log_b n \rfloor}} + a^{\lfloor \log_b n \rfloor - 1} f \left( \frac{n}{\mathbf{t}^{\lfloor \log_b n \rfloor}} \right) + a^{\lfloor \log_b n \rfloor - 1} f \left( \frac{n}{\mathbf{t}^{\lfloor \log_b n \rfloor}} \right) + \dots \right)$$

$$Add_T W_{b \log_b} = \lim_{n \to \infty} t^n p \text{ (p)} w \text{ coder}$$

Continuing in this way  $\log_b n - 1$  many times we get ...

$$\mathbf{h}^{=}_{t} a^{\lfloor \log_b n \rfloor} T / \mathbf{p}^{-n} \mathbf{p}^{-1} \mathbf{v}^{-1} \mathbf{v}^{-1} \mathbf{p}^{-1} \mathbf{f}^{-1} (\mathbf{p}^{-1} \mathbf{p}^{-1} \mathbf{$$

# Add TWeet hat approveder

We now use  $a^{\log_b n} = n^{\log_b a}$ :

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$
 (4)

10/10/12/1

Continuing in this way  $\log_b n - 1$  many times we get ...

$$\mathbf{h}^{=}_{\mathbf{t}} a^{\lfloor \log_b n \rfloor} T \left( \frac{n}{b^{\lfloor \log_b n \rfloor}} + a^{\lfloor \log_b n \rfloor - 1} f \left( \frac{n}{\mathbf{t}^{\lfloor \log_b n \rfloor}} \right) + a^{\lfloor \log_b n \rfloor - 1} f \left( \frac{n}{\mathbf{t}^{\lfloor \log_b n \rfloor}} \right) + \dots \right)$$

# 

We now use  $a^{\log_b n} = n^{\log_b a}$ :

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$
 (4)

Note that so far we did not use any assumptions on f(n). . . f(n)

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$

# Assignment Project Exam Help

https://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assisting in the project A is A in the project A in the project A in the project A is A in the project A in the project A in the project A is A in the project A in the project A in the project A is A in the project A in the project A in the project A is A in the project A in the project A in the project A is A in the project A in the project

https://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assign free $\sum_{i=0}^{\log_b a} a^i f\left(\frac{n}{b^i}\right) = \sum_{i=0}^{\log_b a} a^i O\left(\frac{n}{b^i}\right)^{\log_b a - \varepsilon}$ Exam Help

 $= O(\frac{1}{h} \sum_{i=0}^{\lfloor \log_b p^i \rfloor - 1} S(\frac{\eta}{b^i}) p^i O) wcoder.com$ 

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assisting in the project A is A in the project A is A in the project A is A in the project A

$$= \mathcal{O}\left( \underset{i=0}{\overset{[\log_b n]}{\text{log}_b}^n} \mathbf{S}(\underset{b^i}{\overset{p}{\text{--}}}) \overset{p}{\text{---}} \mathbf{O} \overset{e}{\text{---}} \overset{[\log_b n]}{\text{---}} - \underset{i=0}{\overset{e}{\text{---}}} \mathbf{O} \overset{e}{\text{---}} \overset{i}{\text{----}} \right) \right)$$

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# $Assign f(a) = \sum_{i=0}^{n} a^{i} f\left(\frac{n}{b^{i}}\right) = \sum_{i=0}^{n} a^{i} O\left(\frac{n}{b^{i}}\right)^{\log_{b} a - \varepsilon} Exam Help$

$$= O\left( \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} S\left( \frac{1}{b^i} \right) p^a O \right) w \left( O \left( \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} O \left( \sum_{b=a-\varepsilon}^{i} \right) \right) \right)$$

$$= O \left( \text{Add}^{\log_b a - \varepsilon} \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} \left( \frac{a}{\log_b a + \varepsilon} \right)^i \right)$$

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assisting in the loss of the

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# $\frac{g_{\text{In}}}{g_{a'f}} \underbrace{\text{Project}}_{a'f\left(\frac{n}{b'}\right) = \sum_{a'O\left(\frac{n}{b'}\right)} \log_b a - \varepsilon} \text{Exam Help}$ $= O \left( \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} S(\frac{\eta}{b_i}) \right) O W \left( O d e \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} O(n) \frac{1}{a_b a - \varepsilon} \right)$ $= O\left(n^{\log_b a - \varepsilon} \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} \left(\frac{a}{b^{\log_b a + \varepsilon}}\right)^i\right) = O\left(n^{\log_b a - \varepsilon} \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} \left(\frac{a}{b^{\log_b a + \varepsilon}}\right)^i\right)$ $= O\left(n^{\log_b a - \varepsilon} \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} \left(\frac{a}{a}\right)^i\right)$

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assign The $a^{\log_b a}$ Project Exam Help $\sum_{i=0}^{\infty} a^i f\left(\frac{n}{b^i}\right) = \sum_{i=0}^{\infty} a^i O\left(\frac{n}{b^i}\right)^{\log_b a - \varepsilon}$

$$= O\left( \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} S\left( \frac{\eta}{b^i} \right) p^i O \right) w \left( Ode \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} O(p^i) \frac{1}{a^i - \epsilon} \right) \right)$$

$$=O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1} \left(\frac{a}{b^{\log_b a+\varepsilon}}\right)^i\right) = O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1} \left(\frac{a}{b^{\log_b a-\varepsilon}}\right)^i\right)$$

$$=O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1} \left(\frac{a\,b^\varepsilon}{a}\right)^i\right) = O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1} (b^\varepsilon)^i\right)$$

◆ロト ◆個ト ◆意ト ◆意ト ・ 意 ・ からぐ

$$T(n) \approx n^{\log_b a} T(1) + \underbrace{\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{i=0}$$

# Assign meant project Exam Help $\sum_{i=0}^{\infty} a^{i} f\left(\frac{n}{b^{i}}\right) = \sum_{i=0}^{\infty} a^{i} O\left(\frac{n}{b^{i}}\right)^{\log_{b} a - \varepsilon}$

$$= O\left( \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} S\left( \frac{\eta}{b^i} \right) p^i O\right) w \left( Ode \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} O(n^i) \frac{1}{b^i} \right) \right)$$

$$=O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1}\binom{a}{b^{\log_b a+\varepsilon}}\right)^i = O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1}\binom{a}{b^{\log_b a-\varepsilon}}\right)^i$$

$$=O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1}\left(\frac{a\,b^\varepsilon}{a}\right)^i\right) = O\left(n^{\log_b a-\varepsilon}\sum_{i=0}^{\lfloor\log_b n\rfloor-1}(b^\varepsilon)^i\right)$$

$$=O\left(n^{\log_b a-\varepsilon}\frac{(b^\varepsilon)^{\lfloor \log_b n\rfloor}-1}{b^\varepsilon-1}\right);\quad \text{ we are using } \sum_{i=0}^{m-1}q^i=\frac{q^m-1}{q-1}$$

Case 1 - continued:

$$\underbrace{\sum_{b=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = O\left(n^{\log_b a - \varepsilon} \frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}\right) }_{\text{Assignment Project Exam Help}}$$

https://powcoder.com

Case 1 - continued:

https://powcoder.com

Case 1 - continued:

https://poweoder.com

Case 1 - continued:

$$\underbrace{\sum_{b=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = O\left(n^{\log_b a - \varepsilon} \frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\left(\frac{b^{\varepsilon}}{b^i}\right)^{\lfloor \log_b n \rfloor} - 1}_{b^{\varepsilon} - 1}\right) } \underbrace{\text{Exam Help}}_{}$$

https://posba-e-mosba

Case 1 - continued:

$$\underbrace{\sum_{b=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{(b^{\varepsilon})^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n \rfloor} - 1}\right) }_{=$$

https://posbaeeder.com

Case 1 - continued:

$$\underbrace{\sum_{b=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b n - 1} - 1}\underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\varepsilon}\right)}$$

https://posba-e-missa.e-missa

Since we had:  $T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\infty} t_i \frac{1}{a^i f\left(\frac{n}{b^i}\right)}$  we get:

Case 1 - continued:

$$\underbrace{\sum_{b=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = O\left(n^{\log_b a - \varepsilon} \underbrace{\frac{\left(b^{\varepsilon}\right)^{\lfloor \log_b n \rfloor} - 1}{b^{\varepsilon} - 1}}\right) }_{= O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\ = O\left(n^{\log_b a - \varepsilon} \underbrace{Project}_{b^{\varepsilon} - 1}\right) \\$$

https:  $\sqrt{p_{log_{b}a}^{log_{b}a-\varepsilon}} \sqrt{\frac{n^{log_{b}a-\varepsilon}n^{\varepsilon-1}}{p_{log_{b}a}^{log_{b}a}}} der.com$ 

$$\begin{split} T(n) &\approx n^{\log_b a} T\left(1\right) + O\left(n^{\log_b a}\right) \\ &= \Theta\left(n^{\log_b a}\right) \end{split}$$

◆ロト ◆母ト ◆ヨト ■ りへ○

```
Case 2: f(m) = \Theta(m^{\log_b a})
```

# Assignment Project Exam Help

https://powcoder.com

```
Case 2: f(m) = \Theta(m^{\log_b a})
Assignment = Project Exam Help
```

https://powcoder.com

Case 2: 
$$f(m) = \Theta(m^{\log_b a})$$

Assignment  $e^{-\frac{\log_b n}{n} - 1}$ 

$$= \Theta\left(\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a}\right)$$

https://powcoder.com

Case 2: 
$$f(m) = \Theta(m^{\log_b a})$$

Assignment  $e^{-\log_b n - 1} = e^{\log_b n - 1} e^{-\log_b n - 1} e^{\log_b n -$ 

Case 2: 
$$f(m) = \Theta(m^{\log_b a})$$

Assignment  $e^{\log_b n - 1} = P^{a_i \Theta(n_i) \log_b a} = P^{a_i \Theta(n_i) \log_b a}$ 
 $e^{\log_b n - 1} = P^{a_i \Theta(n_i) \log_b a} = P^{a_i \Theta(n_i)$ 

Case 2: 
$$f(m) = \Theta(m^{\log_b a})$$

Assignment  $e^{-\frac{\log_b n}{\log_b n} - 1} = \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \Theta\left(\frac{n}{b^i}\right)^{\log_b a}$ 
 $e^{-\frac{\log_b n}{\log_b n} - 1} = \Theta\left(\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a}\right)$ 

https://poweroder

Add Wecahat(bgo n) = 1

 $e^{-\frac{\log_b n}{\log_b n} - 1} = \Theta\left(n^{\log_b n} - 1\right)$ 

Case 2: 
$$f(m) = \Theta(m^{\log_b a})$$

Assignment  $e^{\log_b n} = 1$ 
 $e^{\log_b n} = 1$ 

Case 2 (continued):

Assignment Project Exam Help  $\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = \Theta\left(n^{\log_b a} \log_b n\right) = \Theta\left(n^{\log_b a} \log_2 n\right)$ 

https://powcoder.com

Case 2 (continued):

Assignment Project Exam Help

$$\sum_{i=0}^{\log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = \Theta\left(n^{\log_b a} \log_b n\right) = \Theta\left(n^{\log_b a} \log_2 n\right)$$

because long the becaus

$$Add \overset{T(n) \approx n^{\log_b a}T(1) + \sum\limits_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}{WeChat} \overset{\text{log}_b n \rfloor - 1}{powcoder}$$

Case 2 (continued):

Assignment Project Exam Help

$$\sum_{i=0}^{\log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) = \Theta\left(n^{\log_b a} \log_b n\right) = \Theta\left(n^{\log_b a} \log_2 n\right)$$

because long to the long to because long to the long t

 $\text{we get:} \ \, \mathbf{Add} \overset{T(n) \, \approx \, n^{\log_b a} T(1) \, + \, \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{\mathbf{we get:}} \, \mathbf{Add} \overset{T(n) \, \approx \, n^{\log_b a} T(1) \, + \, \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)}_{\mathbf{vecoder}}$ 

$$\begin{split} T(n) &\approx n^{\log_b a} T\left(1\right) + \Theta\left(n^{\log_b a} \log_2 n\right) \\ &= \Theta\left(n^{\log_b a} \log_2 n\right) \end{split}$$



```
Case 3: f(m) = \Omega(m^{\log_b a + \varepsilon}) and a f(n/b) \le c f(n) for some 0 < c < 1.
```

# We get by substitution: $f(n/b) \le \frac{c}{a} f(n)$ Assignment Project Exam Help

 $f(n/b^3) \le \frac{c}{a} f(n/b^2)$ 

https://poweoder.com

Add WeChat powcoder

Case 3:  $f(m) = \Omega(m^{\log_b a + \varepsilon})$  and  $a f(n/b) \le c f(n)$  for some 0 < c < 1.

We get by substitution:

$$f(n/b) \le \frac{c}{a} f(n)$$

## Assignment Project Exam Help

### $f(n/b^3) \le \frac{c}{a} f(n/b^2)$

### https://powcoder.com

By chaining these inequalities we get

$$\mathbf{Add}^{f(n/b^2)} \underbrace{\mathbf{v}}_{f(n/b^2)} \underbrace{\mathbf{v}}_{f(n/b^2)} \underbrace{\mathbf{v}}_{f(n/b^2)} \underbrace{\mathbf{v}}_{f(n/b^2)} \underbrace{\mathbf{v}}_{g(n/b^2)} \underbrace{\mathbf{v}}_{g(n$$

. . .

$$f(n/b^i) \leq \frac{c}{a} \underbrace{f(n/b^{i-1})} \leq \frac{c}{a} \cdot \underbrace{\frac{c^{i-1}}{a^{i-1}} f(n)} = \frac{c^i}{a^i} f(n)$$

Case 3 (continued):

We got 
$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

## Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Case 3 (continued):

We got 
$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

Abssignment Project Exam Help  $\sum_{i=0}^{\log_b n} a^i f\left(\frac{n}{b^i}\right) \leq \sum_{i=0}^{\log_b n} a^i \frac{c^i}{a^i} f(n) < f(n) \sum_{i=0}^{\infty} c^i = \frac{f(n)}{1-c}$ 

$$\sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) \leq \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \frac{c^i}{a^i} f(n) < f(n) \sum_{i=0}^{\infty} c^i = \frac{f(n)}{1-c^i}$$

https://powcoder.com

Add WeChat powcoder

Case 3 (continued):

We got 
$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

Abssignment Project Exam Help  $\sum_{i=0}^{\log_b n} a^i f\left(\frac{n}{b^i}\right) \leq \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \frac{c^i}{a^i} f(n) < f(n) \sum_{i=0}^{\infty} c^i = \frac{f(n)}{1-c}$ 

$$\sum_{i=0}^{\log_b n_1 - 1} a^i f\left(\frac{n}{b^i}\right) \le \sum_{i=0}^{\lfloor \log_b n_1 - 1 \rfloor} a^i \frac{c^i}{a^i} f(n) < f(n) \sum_{i=0}^{\infty} c^i = \frac{f(n)}{1 - c^i}$$

Since we hattps://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$

 $T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$ Add WeChat powcoder

Case 3 (continued):

We got 
$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

Abssignment Project Exam Help  $\sum_{i=0}^{n} a^{i} f\left(\frac{n}{b^{i}}\right) \leq \sum_{i=0}^{\lfloor \log_{b} n \rfloor - 1} a^{i} \frac{c^{i}}{a^{i}} f(n) < f(n) \sum_{i=0}^{\infty} c^{i} = \frac{f(n)}{1-c}$ 

$$\sum_{i=0}^{\log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right) \le \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i \frac{c^i}{a^i} f(n) < f(n) \sum_{i=0}^{\infty} c^i = \frac{f(n)}{1 - c^i}$$

Since we hattps://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$

and since A & do log b We Cet: hat powcoder

$$T(n) < n^{\log_b a} T(1) + O(f(n)) = O(f(n))$$

Case 3 (continued):

$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

## Abssignment Project Exam Help $\sum_{i=0}^{n} a^{i} f\left(\frac{n}{b^{i}}\right) \leq \sum_{i=0}^{\lfloor \log_{b} n \rfloor - 1} a^{i} \frac{c^{i}}{a^{i}} f(n) < f(n) \sum_{i=0}^{\infty} c^{i} = \frac{f(n)}{1-c}$

Since we https://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$

and since A & do log b We Cet: hat powcoder

$$T(n) < n^{\log_b a} T(1) + O(f(n)) = O(f(n))$$

but we also have

$$T(n) = aT(n/b) + f(n) > f(n)$$



Case 3 (continued):

$$f(n/b^i) \le \frac{c^i}{a^i} f(n)$$

## Abssignment Project Exam Help $\sum_{i=0}^{n} a^{i} f\left(\frac{n}{b^{i}}\right) \leq \sum_{i=0}^{\lfloor \log_{b} n \rfloor - 1} a^{i} \frac{c^{i}}{a^{i}} f(n) < f(n) \sum_{i=0}^{\infty} c^{i} = \frac{f(n)}{1-c}$

Since we hattps://powcoder.com

$$T(n) \approx n^{\log_b a} T(1) + \sum_{i=0}^{\lfloor \log_b n \rfloor - 1} a^i f\left(\frac{n}{b^i}\right)$$

and since  $A \in \mathbb{R}^{\log_b}$  We can be a power of the powe

$$T(n) < n^{\log_b a} T(1) + O(f(n)) = O(f(n))$$

but we also have

$$T(n) = aT(n/b) + f(n) > f(n)$$

thus,

$$T(n) = \Theta(f(n))$$

#### Master Theorem Proof: Homework

Exercise 1: Show that condition

## Assignment Project Exam Help

follows from the condition

## https://powcoder.com

**Example:** Let us estimate the asymptotic growth rate of T(n) which satisfies

## Add WeChat-powcoder

**Note:** we have seen that the Master Theorem does **NOT** apply, but the technique used in its proof still works! So let us just unwind the recurrence and sum up the logarithmic overheads.

$$T(n) = 2T\left(\frac{n}{2}\right) + n\log n$$

$$= 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\log\frac{n}{2}\right) + n\log n$$

# $\underbrace{Assignment}^{2^2T\left(\frac{n}{2^2}\right) + n\log\frac{n}{2} + n\log n}_{= 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\log\frac{n}{2^2}\right) + n\log\frac{n}{2} + n\log n}_{= 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\log\frac{n}{2^2}\right) + n\log\frac{n}{2} + n\log n}$

= 23 T ( 1/23) ttp's: 1/2 powcoder.com

$$= 2^{\log n} T\left(\tfrac{n}{2^{\log n}}\right) + n\log\tfrac{n}{2^{\log n-1}} + \ldots + n\log\tfrac{n}{2^2} + n\log\tfrac{n}{2} + n\log n$$

$$= \frac{nT(1)}{A} + \frac{\log 1}{A} \frac{\log 1}{\log 1} = \frac{\log 2}{\ln 1} - \frac{\log 2}{\ln 1} = \frac{\log 2}{\log 1} - \frac{\log 2}{\log 1} = \frac{$$

$$= nT(1) + n((\log n)^2 - \log n(\log n - 1)/2$$

$$= nT(1) + n((\log n)^2/2 + \log n/2)$$

$$=\Theta\left(n(\log n)^2\right).$$

- (ロ) (個) (注) (注) (注) ( ii の)(()

#### **PUZZLE!**

Five pirates have to split 100 bars of gold. They all line up and proceed as follows:

- The first pirate in line gets to propose a way to split up the gold (for example: everyone gets 20 bars)
- The pirates, including the one who proposed, vote on whether to accept the proposal.

  Street pirate in line then makes his proposal, and the 4 pirates vote again. If the vote is tied (2 vs 2) then the proposing pirate is still killed. Only majority can accept a proposal. The process continues until a proposal is accepted or there is only one pirate of the term of that every party if the contraction of the con
  - at of the control of
  - given that he will be alive he wants to get as much gold as possible;
  - given maximal possible amount of gold, he wants to see any other pirate killed, just for fun;
  - Ath of a knowless of t period holder line owcoder
- Question: What proposal should the first pirate make?

Hint: assume first that there are only two pirates, and see what happens. Then assume that there are three pirates and that they have figured out what happens if there were only two pirates and try to see what they would do. Further, assume that there are four pirates and that they have figured out what happens if there were only three pirates, try to see what they would do. Finally assume there are five pirates and that they have figured out what happens if there were only four pirates.