https://pow.coderiveciom

One of the most famous sequences of numbers is the Fibonacci sequence:

Assignment Project Exam Help

https://powcoder.com

One of the most famous sequences of numbers is the Fibonacci sequence:

### Assignment Project Exam Help

The sequence can be defined recursively:

https://powcoder.com  

$$F(n) = \begin{cases} f(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

One of the most famous sequences of numbers is the Fibonacci sequence:

#### Assignment Project Exam Help

The sequence can be defined recursively:

https://powcoder.com  

$$F(n) = \begin{cases} f(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

The Fibonacci numbers have various interesting properties, including Giba related to the guident ratio and its Configuration.

$$\phi = rac{1+\sqrt{5}}{2} pprox 1.61803 \qquad \hat{\phi} = rac{1-\sqrt{5}}{2} pprox -0.61803$$

You could prove by induction that  $F(n) = \frac{\phi^n - \hat{\phi}^n}{\sqrt{5}}$ .

```
Fib(n)

if n = 0 or n = 1

return Fib(n-1) + Fib(n-2)
```

```
Fib(n)

if n = 0 or n = 1

return Fib(n-1) + Fib(n-2)
```

What is the running time P(n) of this provided and input n?

```
Fib(n)

if n = 0 or n = 1

return Fib(n-1) + Fib(n-2)
```

What is the running time P(n) of this provided and input n?

$$T(n) = T(n-1) + T(n-2) + \Theta(1).$$

Assignment Project Exam Help

Assignment Project Exam Help

h t t ps: //powcoder.com

Add WeChat poweoder

Insights:

# Assignment Project Exam Help https://powcoder.com Add WeChat poweroder

#### Insights:

The number of recursive calls at depth i is 2<sup>i</sup>

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

#### Insights:

- The number of recursive calls at depth i is 2<sup>i</sup>
- The shallowest leaf is the leftmost one, with depth  $\frac{n}{2}$ .

### Assignment Project Exam Help

#### Add WeChat powcoder

#### Insights:

- The number of recursive calls at depth i is 2<sup>i</sup>
- The shallowest leaf is the leftmost one, with depth  $\frac{n}{2}$ .

Thus the total number of recursive calls is  $\Omega(2^{\frac{n}{2}})$ , or exponential.

$$T(n) = T(n-1) + T(n-2) + \Theta(1)$$
is 
$$https://powcoder.com$$

$$T(n) = \Theta(\phi^n).$$

is 
$$https://powcoder.com$$

$$T(n) = \Theta(\phi^n).$$

The reason the recursive algorithm is splow is because the same recursive calls are recomputed over and over.

https://powcoder.com

Assignment to a sum be extristed a worth by life p

https://powcoder.com

#### Assignment of a representation of the values returned by the recursive calls.

• Then, at the beginning of each recursive call, check to see if the tips://powerstander.com

#### Assignment and legitisication by the recursive calls.

- Then, at the beginning of each recursive call, check to see if the value we want please exists der.com
- If it does, then we re-use it. Otherwise we compute it.

#### Assignation of the entry legentry legen

- Then, at the beginning of each recursive call, check to see if the value we want please wistoder.com
- If it does, then we re-use it. Otherwise we compute it.

This technique is well-d memorization. powcoder

#### SSI OTTOPO 19 to a septem the ectrsi te ascoption by the ing p the values returned by the recursive calls.

- Then, at the beginning of each recursive call, check to see if thereby we want blood we to der. com
- If it does, then we re-use it. Otherwise we compute it.

- This technique is talled memorization. It:

  maintains the familiar, recursive, the down structure of the algorithm
  - but without the exponential costs of re-computing all the values.

```
// F[0..n] is a global array

MemFib(n)

if https://phewcoder.com

F[n] ← 1

else if F[n] undefined

FA] deminible the fib powcoder

return F[n]
```

```
// F[0..n] is a global array

MemFib(n)

if https://pnewcoder.com

F[n] ← 1

else if F[n] undefined

FA] dem Weight proweder

return F[n]
```

Running Time?

https://powcoder.com

Add WeChat powcoder

```
// F[0..n] is a global array

MemFib(n)

if https://powcoder.com

F[n] ← 1

else if F[n] undefined

F[h] dem Weither bowcoder

return F[n]
```

Running Time? O(n)!

```
// F[0..n] is a global array

MemFib(n)

if https://powcoder.com

F[n] ← 1

else if F[n] undefined

FA] demWhet HemFib(p2)wcoder

return F[n]
```

Running Time? O(n)! An exponential speedup!

Rather than using a top-down approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, recursion and the compute Fibonacci numbers bottom-up, recursion and recursio

https://powcoder.com

Rather than using a top-down approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach inertly a Clause iteratively.

 $\begin{array}{c} \text{To compute } F(n): \\ \text{ } \\$ 



Rather than using a top-down approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach the top top approach the top top approach the top top approach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach the top top approach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach to the recursion and then compute Fibonacci numbers bottom-up, approach to the recursion and then compute Fibonacci numbers bottom-up, approach to the recursion and the recursion and

To compute F(n):

• Metaphy comparing We Grand (a) L. Cap F(1) = 1.



Rather than using a top-down approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, approach iteratively.

To compute F(n):

Multiply comparing We Grand T. Cappil = 1.

• Then we compute F(3) from F(2) and F(1),

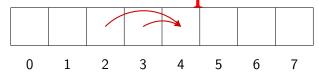


Rather than using a top-down approach to reach the bottom of the recursion and then compute Fibonacci numbers bottom-up, Signification and the recursion and then compute Fibonacci numbers bottom-up, approach file thy a C p iteratively.

To compute F(n):

• Metaphy comparing We Grand (a) L. Cap F(1) = 1.

- Then we compute F(3) from F(2) and F(1),
- · aAthrof We Chaf (2) and so on oder



## Assignment Project Exam Help F[1] ← 1 for i ← 2 to 7 do owcoder.com

#### Add WeChat powcoder

return finl

# Assignment Project Exam Help F[0] ← 1 F[1] ← 1 for i ← 2 to n do Intersi-1/powcoder.com return F[n]

#### Runni And We Chat powcoder

# Assignment Project Exam Help F[1] ← 1 for i ← 2 to n do nttps://powcoder.com return F[n]

#### Runni Atd CIW Chat powcoder

### Assignment Project Exam Help F[1] ← 1 for i ← 2 to 7 do owcoder.com

#### Runni Atd CIW Chat powcoder

return Finl

Note that this algorithm uses  $\Theta(n)$  space. Can you modify it so it uses  $\Theta(1)$  space?

Assignment Project Exam Help

To develop a dynamic algorithm:

https://powcoder.com

To develop a dynamic algorithm:

Pormulate the problem recursively <a href="https://powcoder.com">https://powcoder.com</a>

To develop a dynamic algorithm:

Formulate the problem recursively

To develop a dynamic algorithm:

Formulate the problem recursively

Give a clear recursive formula or algorithm

To develop a dynamic algorithm:

- Formulate the problem recursively

  The the the problem recursively

  Give a clear recursive formula or algorithm
- 2 Build solutions to your recurrence from the bottom up

To develop a dynamic algorithm:

- Formulate the problem recursively

  Give a clear recursive formula or algorithm
- 2 Build solutions to your recurrence from the bottom up

- Formulate the problem recursively

  Give a clear recursive formula or algorithm
- 2 Build solutions to your recurrence from the bottom up
  - A Close a homozation de a trupte WCO der

- Formulate the problem recursively

  Give a clear recursive formula or algorithm
- 2 Build solutions to your recurrence from the bottom up
  - Added a hymeration data truthe WCOder
    - Identify dependencies

### The basic idea behind dynamic programming is recursion without Assignment Project Exam Help

- 1 Formulate the problem recursively
  - The Se the promote Condition of Sold entitlely of Sold entitlely of Sold entitle of Sold entit. The sold entitle of Sold entitle of Sold entitle of Sold entit
- 2 Build solutions to your recurrence from the bottom up
  - A Close a numeration deal truthe WCOCET
    - dentify dependencies
    - Tind a good evaluation order

### The basic idea behind dynamic programming is recursion without Assignment Project Exam Help

- 1 Formulate the problem recursively
  - THE Se the HOLL TO COME COME OF THE PROPERTY O
- 2 Build solutions to your recurrence from the bottom up
  - A Close a numeration deal truthe WCOCET
    - dentify dependencies
    - Tind a good evaluation order
    - Write down the algorithm

### The basic idea behind dynamic programming is recursion without Assignment Project Exam Help

- 1 Formulate the problem recursively
  - THE SE the potential Culant sole entitlely

    Give a clear recursive formula or algorithm
- 2 Build solutions to your recurrence from the bottom up
  - A close a numeration data truttle WCOCET
    - dentify dependencies
    - Find a good evaluation order
    - Write down the algorithm
    - Analyze space and running time

Input: A sequence of characters stored in A[1..n].

Output: True if A can be segmented into a sequence of words,

## Assignment Project Exam Help

https://powcoder.com

Input: A sequence of characters stored in A[1..n].

Output: True if A can be segmented into a sequence of words,

Assignments Work Ojecetur Examuel Lep characters w is a word, False otherwise.

https://powcoder.com

Input: A sequence of characters stored in A[1..n].

Output: True if A can be segmented into a sequence of words,

Assignments Work Ojecetur Examue Help characters w is a word, False otherwise.

Example: If sequence A consists of characters

https://powcoder.com

 $\begin{array}{c} \text{and } \textit{IsWord(w)} \text{ is True if } \textit{w} \text{ is a word in English, then:} \\ Add \ \ \, \text{WeChat powcoder} \end{array}$ 

Input: A sequence of characters stored in A[1..n].

Output: True if A can be segmented into a sequence of words,

Assignments Work Ojecetur Examue Help characters w is a word, False otherwise.

Example: If sequence A consists of characters

https://powcoder.com

Add We Chat powcoder

Output True because

BOTH-EARTH-AND-SATURN-SPIN is a valid segmentation of A

Input: A sequence of characters stored in A[1..n].

Output: True if A can be segmented into a sequence of words,

# Assignments Work Ojecetur Examue Help characters w is a word, False otherwise.

**Example:** If sequence A consists of characters

https://powcoder.com

and IsWord(w) is True if w is a word in English, then:

Add We Chat powcoder

Output True because

BOTH-EARTH-AND-SATURN-SPIN is a valid segmentation of *A* 

 By the way, BOT·HEART·HANDS·AT·URNS·PIN is another valid segmentation of A. Formulate the problem recursively

Assignment Project Exam Help

https://powcoder.com

Formulate the problem recursively

Assignment in formula entire The problem that you want to solve recursively Help

https://powcoder.com

Formulate the problem recursively

Assignment ive formula engrithm X am Help

```
// Is the suffix A[i..n] Splittable?
SplittableS://powcoder.com

if i > n

return True

for A dit We Chat powcoder

if isword(i, y) and splittable of the coder

return True

return False
```

Formulate the problem recursively

Assignment ive formula engrithm Xam Help

```
// Is the suffix A[i..n] Splittable?
SplittableS://powcoder.com

if i > n

return True

for A dit Wechat powcoder

if isword(i, y) can splittable of the coder

return True

return False
```

Running time?

Formulate the problem recursively

Assignment ive formula engrithm Xam Help

```
// Is the suffix A[i..n] Splittable?
SplittableS://powcoder.com

if i > n

return True

for A dit Wechat powcoder

if isword(i, y) can splittable of the coder

return True

return False
```

Running time?  $O(2^n)$ 

https://powcoder.com

## Assignment of the property of

• Only  $O(n^2)$  different ways to call IsWord(i, j), one for each hittips.//powcoder.com

## Assignmentated by the treduse at the positive point Assignment Assignment and <math>Assignment Assignment Assignment and <math>Assignment Assignment As

• Only  $O(n^2)$  different ways to call IsWord(i, j), one for each the such that the computing only a polynomial amount of stuff???"

## Assignment of the property of

• Only  $O(n^2)$  different ways to call IsWord(i, j), one for each the such that the computing only a polynomial amount of stuff???"

### For Medd WeChat powcoder

BLUE STEM UNIT ROBOT HEARTHANDSATURNSPIN

BLUEST EMU NITRO BOT HEARTHANDSATURNSPIN

Build solutions to your recurrence from the bottom up

- a Each recursive subproblem is Splittable(i) with i between
- Assignment Project Exam Help

https://powcoder.com

Build solutions to your recurrence from the bottom up

6 Each recursive subproblem is Splittable(i) with i between

Assignment of Project pleasant a Help

SplitTable[1..n+1].

https://powcoder.com

Build solutions to your recurrence from the bottom up

- a Each recursive subproblem is Splittable(i) with i between
- AssignmentoiProjectplEtxamaHelp
  - Each subproblem Splittable(i) depends only on results of hopes splittable(i) cwherd in icom



Build solutions to your recurrence from the bottom up

- a Each recursive subproblem is Splittable(i) with i between
- Assignmentoie traject pletzam Help

  SplitTable[1..n+1].
  - Each subproblem Splittable(i) depends only on results of horosphems Splittable(i) where it is com



d ... so we should be filling the array in decreasing index order, starting with SplitTable[n+1] = True.

Build solutions to your recurrence from the bottom up

- ${f a}$  Each recursive subproblem is Splittable(i) with i between
- AssignmentoiProjectplEtxamaHelp
  - Each subproblem Splittable(i) depends only on results of horosphems Splittable(i) where it is com



- d ... so we should be filling the array in decreasing index order, starting with SplitTable[n+1] = True.
- The algorithm is on the next slide ...
- 1 Also on next slide, the running time analysis ...

### Text segmentation dynamic programming algorithm

```
A significant Project Exam Help

Fast Plittable (A[1..n]):

SplitTable [n + 1] 	True

for i 	n down to 1

altip [i] polycoder.com

for j 	i to n

if IsWord(i, j) and SplitTable [j + 1]

A SplitTable [i] Crue

return splitTable [i] Chat powcoder
```

### Text segmentation dynamic programming algorithm

```
A significant Project Exam Help

Fast split table (A[1..n]):

Split Table [n + 1] ← True

for i ← n down to 1

Split Table [i] Palw Coder.com

for j ← i to n

if Is Word(i, j) and Split Table [j + 1]

A Split Table [i] Cruet powcoder

return split Table [i] Chat powcoder
```

Running time?

### Text segmentation dynamic programming algorithm

```
Assignments project Exam Help

Fast splittable (A[1..n]):

SplitTable [n + 1] ← True

for i ← n down to 1

SplitTable [i] Poly Coder.com

for j ← i to n

if IsWord(i, j) and SplitTable [j + 1]

AsplitTable [i] Chat powcoder

return splitTable [i] Chat powcoder
```

Running time?  $O(n^2)$ 

For any sequence S,

Assignment Project Exam Help

https://powcoder.com

#### Longest Increasing Subsequence

For any sequence S, a subsequence of S is another sequence obtained from S by deleting zero or more elements, without changing the order of the remaining elements.

SSIGNMENT PROJECT EXAM Help

https://powcoder.com

### Longest Increasing Subsequence

For any sequence S, a subsequence of S is another sequence obtained from S by deleting zero or more elements, without changing the order of the remaining elements.

SSISHMENT PROJECT EXAMELED

Problem: Given a sequence of integers S find the Longest Increasing Space of the Communication of the Communicatio

#### Longest Increasing Subsequence

For any sequence S, a subsequence of S is another sequence obtained from S by deleting zero or more elements, without

changing the order of the remaining elements.

ASSIGNMENT Project Exam Help

Problem: Given a sequence of integers S find the Longest Increasing Subsquencial WEOGER.COM

Input: Integer array A[1..n].

Output: Longest possible sequence of indices

Add We Chat powcoder

Example: See above.

Formulate the problem recursively

a Describe the problem that you want to solve recursively

## Assignment Project Exam Help

https://powcoder.com

Formulate the problem recursively

a Describe the problem that you want to solve recursively

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Formulate the problem recursively

a Describe the problem that you want to solve recursively

## Assignment Project Exam Help

```
// return length of LIS of A[j..n] s.t.
// every element, is larger than 4[i]
                powcoder.com
   return 0
 else if A[i] We Chat powcoder
 else
   skip \leftarrow LISbigger(i, j + 1)
   take \leftarrow LISbigger(j, j + 1) + 1
   return max{skip, take}
```

Formulate the problem recursively

a Describe the problem that you want to solve recursively

### Assignment Project Exam Help

```
// return length of LIS of A[j..n] s.t.
// every element is larger than A[i]
LISb net 0.Si)//powcoder.com
    return 0
  else if A[i] We Chat powcoder
  else
    skip \leftarrow LISbigger(i, j + 1)
    take \leftarrow LISbigger(j, j + 1) + 1
    return max{skip, take}
```

Formulate the problem recursively

a Describe the problem that you want to solve recursively

## Assignment Project Exam Help

```
// return length of LIS of A[j..n] s.t.
// every element is larger than A[i] LISb net p.Si)./powcoder.com
    return 0
  else if A[i] We Chat powcoder
  else
    skip ← LISbigger(i, j + 1)
    take \leftarrow LISbigger(j, j + 1) + 1
    return max{skip, take}
```

### Assignmental Project Exam Help

- a Describe the problem that you want to solve recursively
- Give a clear recursive formula or algorithm

  Plants of Company to Company to
  - a Identify the subproblems
  - Choose a memoization data structure
  - A cid good even tion hat powcoder
    - Write down the algorithm
    - **f** Analyze space and running time

Build solutions to your recurrence from the bottom up

Assimptoblems are Disbigger (i.t.) F1) and Help between 0 and n

https://powcoder.com

Add WeChat powcoder

Build solutions to your recurrence from the bottom up

Assimproblems are Disbigger (i.t. j F.t.) and Help between 0 and n

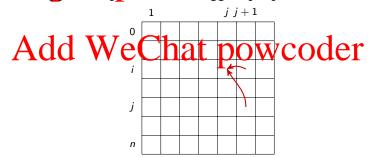
We can memoize the results of these subproblems into a two differsional and two differsional areas and two differsional areas are the differsional areas are two differsional areas are the differsional areas are the differsional areas are the difference are the difference areas are the difference are the

### Add WeChat powcoder

Build solutions to your recurrence from the bottom up

- The subproblems are LISbigger(i, j + 1) and

  LISbigger(i, j + 1) with indicate i and invite value.
- Assignment Project Exam Help
  - **b** We can memoize the results of these subproblems into a two-dimensional array LISbigger [0..n, 1..n+1].
  - fach entry LISbigger [i, j] is filled in using entries LISbigger [i, j] and LISbigger [j, j+1].



Build solutions to your recurrence from the bottom up

Assimption of the subproblems are Disbigger (it j F 1) and Help between 0 and n

- We can memoize the results of these subproblems into a the dip sonal and we get the continuous forms of the can memoize the results of these subproblems into a the dip sonal and the can memoize the results of these subproblems into a the dip sonal and the can memoize the results of these subproblems into a the dip sonal and the can memoize the results of these subproblems into a the dip sonal and the can memoize the results of these subproblems into a the dip sonal and the can memoize the results of these subproblems into a the dip sonal and the can memoize the can me
- Each entry LISbigger[i, j] is filled in using entries LISbigger[i, j+1] and LISbigger[j, j+1].
- FAir he entitle of the the giften polyther column-by-column, right-to-left.

Build solutions to your recurrence from the bottom up

# Assimptoblems are Disbigger (it j F 1) and Help between 0 and n

- We can memoize the results of these subproblems into a the dip sonal and we get the continuous forms of the continuous forms of these subproblems into a the dip sonal and the continuous forms of the
- Each entry LISbigger[i, j] is filled in using entries LISbigger[i, j+1] and LISbigger[j, j+1].
- Fixing he entitle with the time interpolytic entitle column by-column, right-to-left.
- Iterative algorithm on the slide...
- Running time analysis on the next slide...

```
// return length of LIS of A[1..n]
FastLIS(A[1 .. n]):
szignment Project Exam Help
   LISbigger[i, n + 1] \leftarrow 0
 for j \leftarrow n down to 1
   https://powcoder.com
    keep ← 1 + LTSbigger[j, j + 1]
    skip ← LISbigger[i, j + 1]
          wechat powcoder
    else
      LISbigger[i, j] \leftarrow max{keep, skip}
 return LISbigger[0, 1]
```

```
// return length of LIS of A[1..n]
FastLIS(A[1 .. n]):
ssignment Project Exam Help
   LISbigger[i, n + 1] \leftarrow 0
 for j \leftarrow n down to 1
   https://powcoder.com
    keep \leftarrow 1 + LTSbigger[j, j + 1]
    skip ← LISbigger[i, j + 1]
           wechat powcoder
     else
      LISbigger[i, j] \leftarrow max{keep, skip}
 return LISbigger[0, 1]
```

```
// return length of LIS of A[1..n]
FastLIS(A[1 .. n]):
ssignment Project Exam Help
   LISbigger[i, n + 1] \leftarrow 0
 for j to down to 1 powcoder.com
    keep ← 1 + LTSbigger[j, j + 1]
    skip ← LISbigger[i, j + 1]
          ggWieChat powcoder
    else
      LISbigger[i, j] \leftarrow max{keep, skip}
 return LISbigger[0, 1]
```