A paper with text and numbers

Description automatically generated

A questionnaire with text and images

Description automatically generated with medium confidence

Embedded file printout 10_607_Homework_4___Latex_Template_3.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
1 Recursive Sequences (6 points) 
Take the recursive sequence defined here: 
f(n) = n,V0 S 3 
10-607 
Download . zip and implement sequencel, sequence2, and sequence3 in 
the file hw4 . py. 
Implement this sequence recursively in sequencel ( ) . 
Implement the sequence iteratively in sequence 2 ( ) . 
Implement the sequence recursively with memoization in sequence3 ( ) . 
For each of the questions, you need to fill in the function bodies (in place of YOUR CODE HERE). 
To test your code locally on a subset of Gradescope tests, run the script hw4_t s. py. After you are 
done, you will submit your code to Gradescope, where we wil run your code on a full suite of tests and 
the autograder Will assign your code points on whether it passes the tests. 
We also provided a file sequence_time . py that contains code for you to observe the speed Of the 
techniques you have implemented for the sequence. Run the script and take a minute to observe and 
reason about the relative amount Of time each technique uses. 
l. (2 points) Short answer: Find the time complexity of the naive recursive implementation of the se- 
quence in sequencel ( ) . Observe that it is very bad, and explain why that is the case based on your 
Solution 
2. (2 points) Short answer: Find the time complexity of your iterative implementation of the sequence 
(sequence2 ( ) ) based on n. 
Solution 
3. (2 points) Short amswer: Find the time complexity of your recursive implementation with memoim- 
tion (sequence3 ( ) ) based on n. 
3 of7 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Embedded file printout 10_607_Homework_4___Latex_Template_3.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
1 Recursive Sequences (6 points) 
Take the recursive sequence defined here: 
f(n) = n,V0 S 3 
10-607 
Download . zip and implement sequencel, sequence2, and sequence3 in 
the file hw4 . py. 
Implement this sequence recursively in sequencel ( ) . 
Implement the sequence iteratively in sequence 2 ( ) . 
Implement the sequence recursively with memoization in sequence3 ( ) . 
For each of the questions, you need to fill in the function bodies (in place of YOUR CODE HERE). 
To test your code locally on a subset of Gradescope tests, run the script hw4_t s. py. After you are 
done, you will submit your code to Gradescope, where we wil run your code on a full suite of tests and 
the autograder Will assign your code points on whether it passes the tests. 
We also provided a file sequence_time . py that contains code for you to observe the speed Of the 
techniques you have implemented for the sequence. Run the script and take a minute to observe and 
reason about the relative amount Of time each technique uses. 
l. (2 points) Short answer: Find the time complexity of the naive recursive implementation of the se- 
quence in sequencel ( ) . Observe that it is very bad, and explain why that is the case based on your 
Solution 
2. (2 points) Short answer: Find the time complexity of your iterative implementation of the sequence 
(sequence2 ( ) ) based on n. 
Solution 
3. (2 points) Short amswer: Find the time complexity of your recursive implementation with memoim- 
tion (sequence3 ( ) ) based on n. 
3 of7 


Embedded file printout 10_607_Homework_4___Latex_Template_4.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
Solution 
4 of7 
10-607 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Embedded file printout 10_607_Homework_4___Latex_Template_4.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
Solution 
4 of7 
10-607 


Embedded file printout 10_607_Homework_4___Latex_Template_5.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
2 Recursion (2 points) 
10-607 
Download 10607_f23 _hw4 . zip and implement the following questions in the file hw4 . py. 
Implement the functions recursive_solution (left hand side) and static_solution 
(right hand side) Of the following equation: 
i(i + l) 
• In Python, we commonly use the Ten ( ) function to determine the length Of a string. Turns out, 
recursion is a viable way to find the length or a string. Implement the function 
Without using Python's built in Ten ( ) . You Will not earn any points if you use Python's built in 
function. 
• The sum of a finite geometric series is defined as 
= aro 4- • 4- arn¯l 
Implement the sum both recursively in geometri c_sum (left hand side) and statically in 
geomet n (right hand side). 
For each Of the questions, you need to in the function bodies (in place Of YOUR CODE HERE). 
TO test your code locally on a subset or Grade-scope tests, run the script hw4_tests . py. After you are 
done, you will submit your code to Gradescope, where we wil run your code on a full suite of tests and 
the autograder will assign your code points based on whether it passes the tests. 
l. (2 points) If you were implementing the Python library, how would you go about implementing the 
len ( ) function for a string? Argue for your solution based on its computational complexity and space 
complexity. 
Solution 
5 of7 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Embedded file printout 10_607_Homework_4___Latex_Template_5.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
2 Recursion (2 points) 
10-607 
Download 10607_f23 _hw4 . zip and implement the following questions in the file hw4 . py. 
Implement the functions recursive_solution (left hand side) and static_solution 
(right hand side) Of the following equation: 
i(i + l) 
• In Python, we commonly use the Ten ( ) function to determine the length Of a string. Turns out, 
recursion is a viable way to find the length or a string. Implement the function 
Without using Python's built in Ten ( ) . You Will not earn any points if you use Python's built in 
function. 
• The sum of a finite geometric series is defined as 
= aro 4- • 4- arn¯l 
Implement the sum both recursively in geometri c_sum (left hand side) and statically in 
geomet n (right hand side). 
For each Of the questions, you need to in the function bodies (in place Of YOUR CODE HERE). 
TO test your code locally on a subset or Grade-scope tests, run the script hw4_tests . py. After you are 
done, you will submit your code to Gradescope, where we wil run your code on a full suite of tests and 
the autograder will assign your code points based on whether it passes the tests. 
l. (2 points) If you were implementing the Python library, how would you go about implementing the 
len ( ) function for a string? Argue for your solution based on its computational complexity and space 
complexity. 
Solution 
5 of7 


Embedded file printout 10_607_Homework_4___Latex_Template_6.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
10-607 
3 
Code to Equation (6 points) 
Given each recursive function, write the corresponding function mathematically. 
(3 points) Short answer: Function f where is an integer. Assume n l. Write an expression for 
def f (n): 
if n — 
return 2/5 
return 2 * n / 
Solution 
2. (3 points) Short answer: Function g where n is an integer. Assume n 0. Write an expression for 
f(n). 
def g (n) : 
if O n 3: 
return 
if n 3 
return 
elif n 2 
return 
else: 
return 
Solution 
g(n-l) 
+ g (n—3) 
6 of7 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Embedded file printout 10_607_Homework_4___Latex_Template_6.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
10-607 
3 
Code to Equation (6 points) 
Given each recursive function, write the corresponding function mathematically. 
(3 points) Short answer: Function f where is an integer. Assume n l. Write an expression for 
def f (n): 
if n — 
return 2/5 
return 2 * n / 
Solution 
2. (3 points) Short answer: Function g where n is an integer. Assume n 0. Write an expression for 
f(n). 
def g (n) : 
if O n 3: 
return 
if n 3 
return 
elif n 2 
return 
else: 
return 
Solution 
g(n-l) 
+ g (n—3) 
6 of7 


Embedded file printout 10_607_Homework_4___Latex_Template_7.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
4 Collaboration Questions 
10-607 
After you have completed all other components of this assignment, report your answers to these questions 
regarding the collaboration policy. Details of the policy can be found in the syllabus. 
1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full 
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details. 
3. Did you find or come across code that implements any part Of this assignment? If so, include full 
details. 
Your Answer 
7 of7 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Embedded file printout 10_607_Homework_4___Latex_Template_7.pdf Machine generated alternative text:
Homework 4: Recursion and Dynamic Programming 
4 Collaboration Questions 
10-607 
After you have completed all other components of this assignment, report your answers to these questions 
regarding the collaboration policy. Details of the policy can be found in the syllabus. 
1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full 
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details. 
3. Did you find or come across code that implements any part Of this assignment? If so, include full 
details. 
Your Answer 
7 of7 
