A3 Rubric

NOTE: All rubric columns are provided as guidelines for the marks you can expect. In reality, every rubric item can receive any mark within 0.5 increments from 0 to the maximum possible mark.

Common mistakes are listed for you to check against your code, but not all are weighted equally.

Code Submission - 50 Marks

	Poor Performance	Weak Performance	Average Performance	Good Performance	Great Performance	Total Marks
Task 1 - An Island a Day...						30
Approach **Common Mistakes** Inefficient but correct approach Overcomplicates solution	No attempt made, or extremely poor attempt that doesn't demonstrate any knowledge of the task or it's solution [0 Marks]	Some knowledge of the requirements are demonstrated but solution fails at achieving some functionality and is inefficient. [3 Marks]	An average attempt. Solves the problem, but is likely inefficient, or simplifies the problem a bit too much, causing logical issues. [6 Marks]	The solution given is functional, and solves all problems the task puts forward. There are some minor issues with efficiency, but for the most part knowledge of the topics at hand are demonstrated. [9 Marks]	Perfect Approach, Efficient, and uses the concepts taught within the unit [12 Marks]	
Tests **Common Mistakes** Failing the provided test cases Failing tests provided by instructors	Passes barely any tests [0 Marks]	Passes a good amount of the provided tests [3 Marks]	Solves all provided tests, but fails most of the hidden tests, and does so without hardcoding answers [6 Marks]	Failes some tests that the instructor can provide, but also passes many of the hidden tests [9 Marks]	Passes all tests, code is very robust. [12 Marks]	
Documentation **Common Mistakes** Incorrect complexity analysis given Ignores affects of duplicating python objects / assigning values Ignores hidden affects of data structure methods Analysis does not state complexity in terms of well defined variables Best case given assumes small input size **Missing explanatory paragraph** **Explanatory paragraph missing mention of ADTs, short example, etc.**	Analysis given for all required functions is missing, or functions not attempted. [0 Marks]	Somewhere inbetween the two adjacent descriptions. [1.5 Marks]	Analysis for functions given is correct most of the time, and since the task has been adequately attempted this analysis is non-trivial. Some explanation of approach is given. [3 Marks]	Somewhere inbetween the two adjacent descriptions. [4.5 Marks]	Analysis for all functions is correct. Correct variables are used and where required, extra argumentation is given. Short paragraph included explaining approach, mentioning ADTs used, and a short example. [6 Marks]	
Task 2 - Davy Back Fight						20
Approach **Common Mistakes** Inefficient but correct approach Overcomplicates solution	No attempt made, or extremely poor attempt that doesn't demonstrate any knowledge of the task or it's solution [0 Marks]	Some knowledge of the requirements are demonstrated but solution fails at achieving some functionality and is inefficient. [2 Marks]	An average attempt. Solves the problem, but is likely inefficient, or simplifies the problem a bit too much, causing logical issues. [4 Marks]	The solution given is functional, and solves all problems the task puts forward. There are some minor issues with efficiency, but for the most part knowledge of the topics at hand are demonstrated. [6 Marks]	Perfect Approach, Efficient, and uses the concepts taught within the unit [8 Marks]	
Tests **Common Mistakes** Failing the provided test cases Failing tests provided by instructors	Passes barely any tests [0 Marks]	Passes a good amount of the provided tests [2.5 Marks]	Solves all provided tests, but fails most of the hidden tests, and does so without hardcoding answers [3 Marks]	Failes some tests that the instructor can provide, but also passes many of the hidden tests [4.5 Marks]	Passes all tests, code is very robust. [6 Marks]	
Complexity Analysis **Common Mistakes** Incorrect complexity analysis given Ignores affects of duplicating python objects / assigning values Ignores hidden affects of data structure methods Analysis does not state complexity in terms of well defined variables Best case given assumes small input size	Analysis given for all required functions is missing, or functions not attempted. [0 Marks]	Somewhere inbetween the two adjacent descriptions. [1.5 Marks]	Analysis for functions given is correct most of the time, and since the task has been adequately attempted this analysis is non-trivial. Some explanation of approach is given. [3 Marks]	Somewhere inbetween the two adjacent descriptions. [4.5 Marks]	Analysis for all functions is correct. Correct variables are used and where required, extra argumentation is given. Short paragraph included explaining approach, mentioning ADTs used, and a short example. [6 Marks]	
Penalties						50
Formatting Penalty **Common Mistakes** Code not submitted through Ed. Files renamed, or corrupt files uploaded. Some scaffold files removed	Some or all of the common mistakes made. Anywhere from -1 to -5 Marks				No issues [0 Marks]	
Late Penalty **Common Mistakes** Submitting Late	7 Days Late [-35 Marks]	Submitted 5 Days Late [-25 Marks]	Submitted 3 Days Late [-15 Marks]	Submitted 1 Day Late [-5 Marks]	Submitted on time [0 Marks]	

Theory Assessment - 50 Marks