Marks distribution [250]

Marks are uploaded on moodle out of **100. Scale your 250 marks according to the rubric into 100 by the formula: (marks obtained/250)\*100**

Demo [25 marks]

Question-1 [110 Marks]

Question-2 [115 Marks]

General Note Demo:

1. TAs could ask any question related to assignment. Following are few suggested questions

What is the space complexity of program power(l)?

How to handle a set of sets input (list of lists)?

1. For any question test cases carry **60%** marks and **40%** percent marks equally distributed to time complexity, loop invariant, representation invariant, representative test data. Not all questions require all of these analyses.

Eg: isEmpty don’t require loop invariant and representation invariant so 40% marks are distributed for time complexity and representative test data.

General Note Q1:

1. Test Cases and Correctness: [60% marks]

Each question has different weightage for test cases. Each test case in question have equal weightage. Note down the test case score from the test script output.

1. Loop Invariants:

Make sure that students have written the correct loop invariant for required questions.

Each question has a different weightage of loop invariants. You can do partial markings if you think analysis is incomplete.

1. Representation Invariant:

Few questions required representation invariant analysis. Make sure that they have assumed representation invariant for input and formally/informally proved that output also follows representation invariant. In first question representation invariant should be **“List does not contain any duplicate elements”**

1. Representative Test Data:

Make sure that students have provided at least one example test data per question which they used for checking their code.

The test date can either be in comment or print statement with expected outcome in comment.

1. Time Complexity:

Time complexity is supposed to be analyzed for each of the parts, make sure they are correct and students are able to explain.

In part 1 we are not looking for efficient algorithms so even if time complexity of algorithms is higher than expected do not deduce marks.

1. Penalties:
2. If students use **if x in L:** in a function then give them **0 mark** for that question.

Marks Distribution for Q1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | Test cases | Time Complexity | Loop invariant | Representation invariant | Representative test data | Total |
| emptySet | 0.6 | 0.4 | 0 | 0 | 0 | 1 |
| isEmpty | 1.2 | 0.4 | 0 | 0 | 0.4 | 2 |
| member | 6 | 1.33 | 1.33 | 0 | 1.33 | 10 |
| singleton | 1.2 | 0.4 | 0 | 0 | 0.4 | 2 |
| isSubset | 9 | 2 | 2 | 0 | 2 | 15 |
| setEqual | 9 | 2 | 2 | 0 | 2 | 15 |
| union | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| intersection | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| cartesian | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| power | 12 | 2 | 2 | 2 | 2 | 20 |

Test Cases:

1.1 emptyset(): [1 mark]

TestCase: (0.6)

emptyset() = []

Time Complexity: (0.4)

O(1)

1.2 isEmpty(S): [2 mark]

TestCases: (1.2)

isEmpty([[]) = True

isEmpty([1,2,4]) = False

Time Complexity: (0.4)

O(1)

Representative Test Data: (0.4)

1.3 member(S,e): [10 mark]

TestCases: (6)

member([],2) = False

member([1,2,3],2) = True

member([1,2,3],4) = False

member(["col","mtl"],"COL") = False

member(["Alan turing","Tim Berners-Lee"],"Alan turing") = True

member([(4,"mtl"),(4,"col"),(2,"cmp")],(100000,"col")) = False

member([(4,"mtl"),(4,"col"),(2,"cmp")], ,(4,"col")) = True

Time Complexity: (1.33)

O(n)

Loop invariant: (1.33)

Representative Test Data: (1.33)

1.4 singleton(x): [2 mark]

TestCases: (1.2)

singleton(0.0) = [0.0]

singleton(23) = [23]

singleton('a') = ['a']

singleton("Alan Turing") = ["Alan Turing"]

singleton((2,"cmp")) = [(2,"cmp")]

Time Complexity: (0.4)

O(1)

Representative Test Data: (0.4)

1.5 isSubset(P,Q): [15 mark]

TestCases: (9)

isSubset(['Arjun', 'Karn'],['Yudhishthir', 'Bhim', 'Arjun', 'Nakul', 'Sahadev']) = False

isSubset([],['Yudhishthir', 'Bhim', 'Arjun', 'Nakul', 'Sahadev']) = True

isSubset([2],[1, 2, 3, 5, 8]) = True

isSubset([(66, 'Belphegor')],[(0, 'Aryabhatta '), (1729, 'Ramanujan')]) = False

isSubset([(0, 'Aryabhatta ')],[(0, 'Aryabhatta '), (1729, 'Ramanujan')]) = True

Time Complexity: (2)

O(n\*m)

Loop invariant: (2)

Representative Test Data: (2)

1.6 setEqual(P,Q): [15 mark]

TestCases: (9)

setEqual(['virat', 'dhoni', 'sachin', 'rohit'],['scahin', 'dhoni', 'virat', 'rohit']) = True

setEqual([1.0, -1.0],[-1.0, 1.0]) = True

setEqual([],['its empty']) = False

Time Complexity: (2)

O(n\*m)

Loop invariant: (2)

Representative Test Data: (2)

1.7 union(P,Q): [15 mark]

TestCases: (9)

intersection(['delhi', 'mumbai', 'banglore'],['delhi', 'kolkata']) = ['delhi']

intersection([1.0, -1.0, 2.0, 3.0],[-1.0, 1.0]) = [1.0, -1.0]

intersection([],['its empty']) = []

Time Complexity: (1.5)

O(n\*m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

1.8 intersection(P,Q): [15 mark]

TestCases: (9)

Time Complexity: (1.5)

O(n\*m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

1.9 cartesian(P,Q): [15 mark]

TestCases: (9)

cartesian([1, 2, 3],[1]) = [(1, 1), (2, 1), (3, 1)]

cartesian([2, 1],['a', 'b']) = [(2, 'a'), (1, 'a'), (1, 'b'), (2, 'b')]

cartesian([],['its empty']) = []

Time Complexity: (1.5)

O(n\*m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

1.10 power(P): [20 mark]

TestCases: (12)

power([1, 2, 3]) = [[], [1], [2], [3], [1, 2], [1, 3], [3, 2], [1, 2, 3]]

power([2, 1]) = [[], [2], [1], [1, 2]]

power([]) = [[]]

Time Complexity: (2)

O()

Loop invariant: (2)

Representative Test Data: (2)

Representation Invariant: (2)

General Note Q2:

1. Test Cases and Correctness: [60% marks]
2. Each question has different weightage for test cases. Each test case in question has equal weightage. Note down the test case score from the test script output..
3. If students have used sorting, award them **0 marks.**
4. Students are allowed to use the ‘==’ operator**.**
5. Loop Invariants:

Make sure that students have written the correct loop invariant for required questions.

Each question has a different weightage of loop invariants.

1. Representation Invariant:

Few questions required representation invariant analysis. Make sure that they have assumed representation invariant for input and formally/informally proved that output also follows representation invariant. In this question representation invariant should be **“List does not contain any duplicate elements and elements are in sorted order ”**

1. Representative Test Data:

Make sure that students have provided at least one example test data per question which they used for checking their code.

1. Time Complexity:

Time complexity is supposed to be analyzed for each of the parts, make sure they are correct and students are able to explain.

1. Penalties:
2. If students use the same logic as part 1 then give them **0 marks.**
3. If a student has used an algorithm which has O(nlog(n)), O(mlog(n)) type complexity for questions instead of the expected linear algorithm, give them a **50% penalty.**
4. If students use **if x in L:** in a function then give them **0 mark** for that question.

Marks Distribution for Q2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | Test cases | Time Complexity | Loop invariant | Representation invariant | Representative test data | Total |
| emptySet\_2 | 0.6 | 0.4 | 0 | 0 | 0 | 1 |
| isEmpty\_2 | 1.2 | 0.4 | 0 | 0 | 0.4 | 2 |
| member\_2 | 9 | 2 | 2 | 0 | 2 | 15 |
| singleton\_2 | 1.2 | 0.4 | 0 | 0 | 0.4 | 2 |
| isSubset\_2 | 9 | 2 | 2 | 0 | 2 | 15 |
| setEqual\_2 | 9 | 2 | 2 | 0 | 2 | 15 |
| union\_2 | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| intersection\_2 | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| cartesian\_2 | 9 | 1.5 | 1.5 | 1.5 | 1.5 | 15 |
| power\_2 | 12 | 2 | 2 | 2 | 2 | 20 |

Test Cases:

2.1 emptyset\_2(): [1 mark]

TestCase: (0.6)

emptyset\_2() = []

Time Complexity: (0.4)

O(1)

2.2 isEmpty\_2(S): [2 mark]

TestCases: (1.2)

isEmpty\_2([]) = True

isEmpty\_2([1, 2, 4]) = False

isEmpty\_2(['A', 'B']) = False

isEmpty\_2(['a', 'c']) = False

Time Complexity: (0.4)

O(1)

Representative Test Data: (0.4)

2.3 member\_2(S,e): [15 mark]

TestCases: (9)

member\_2([],2) = False

member\_2([1, 2, 3],2) = True

member\_2([1, 2, 3],4) = False

member\_2(['col', 'mtl'],COL) = False

member\_2(['Alan turing', 'Tim Berners-Lee'],Alan turing) = True

member\_2([(2, 'cmp'), (4, 'col'), (4, 'mtl')],(100000, 'col')) = False

member\_2([(2, 'cmp'), (4, 'col'), (4, 'mtl')],(4, 'col')) = True

Time Complexity: (2)

O(log(n))

Loop invariant: (2)

Representative Test Data: (2)

2.4 singleton\_2(x): [2 mark]

TestCases: (1.2)

singleton\_2(0.0) = [0.0]

singleton\_2(23) = [23]

singleton\_2(a) = ['a']

singleton\_2(Alan Turing) = ['Alan Turing']

singleton\_2((2, 'cmp')) = [(2, 'cmp')]

Time Complexity: (0.4)

O(1)

Representative Test Data: (0.4)

2.5 isSubset\_2(P,Q): [15 mark]

TestCases: (9)

isSubset\_2(['Arjun', 'Karn'],['Arjun', 'Bhim', 'Nakul', 'Sahadev', 'Yudhishthir']) = False

isSubset\_2([],['Arjun', 'Bhim', 'Nakul', 'Sahadev', 'Yudhishthir']) = True

isSubset\_2([2],[1, 2, 3, 5, 8]) = True

isSubset\_2([(66, 'Belphegor')],[(0, 'Aryabhatta '), (1729, 'Ramanujan')]) = False

isSubset\_2([(0, 'Aryabhatta '), (1729, 'Ramanujan')],[(0, 'Aryabhatta '), (1729, 'Ramanujan')]) = True

Time Complexity: (1.5)

O(n+m) or O(nlog(m))

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

2.6 setEqual\_2(P,Q): [15 mark]

TestCases: (9)

setEqual\_2(['dhoni', 'rohit', 'sachin', 'virat'],['dhoni', 'rohit', 'sachin']) = False

setEqual\_2([-1.0, 1.0],[-1.0, 1.0]) = True

setEqual\_2([],['its empty']) = False

Time Complexity: (2)

O(min(m,n)) or O(m) or O(n)

Loop invariant: (2)

Representative Test Data: (2)

2.7 union\_2(P,Q): [15 mark]

TestCases: (9)

union\_2(['sachin', 'virat'],['bumrah', 'dhoni']) = ['bumrah', 'dhoni', 'sachin', 'virat']

union\_2([-1.0, 1.0, 2.0, 3.0],[-1.0, 1.0]) = [-1.0, 1.0, 2.0, 3.0]

union\_2([],['its empty']) = ['its empty']

Time Complexity: (1.5)

O(n+m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

2.8 intersection\_2(P,Q): [15 mark]

TestCases: (9)

intersection\_2(['banglore', 'delhi', 'mumbai'],['delhi', 'kolkata']) = ['delhi']

intersection\_2([-1.0, 1.0, 2.0, 3.0],[-1.0, 1.0]) = [-1.0, 1.0]

intersection\_2([],['its empty']) = []

Time Complexity: (1.5)

O(n+m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

2.9 cartesian\_2(P,Q): [15 mark]

TestCases: (9)

cartesian\_2([1, 2, 3],[1]) = [(1, 1), (2, 1), (3, 1)]

cartesian\_2([1, 2],['a', 'b']) = [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]

cartesian\_2([],['its empty']) = []

Time Complexity: (1.5)

O(n\*m)

Loop invariant: (1.5)

Representative Test Data: (1.5)

Representation Invariant: (1.5)

2.10 power\_2(P): [20 mark]

TestCases: (12)

power\_2([1, 2, 3]) = [[], [1], [1, 2], [1, 2, 3], [1, 3], [2],[2,3],[3]]

power\_2([1,2]) = [[], [1], [1, 2], [2]]

power\_2([]) = [[]]

Time Complexity: (2)

O()

Loop invariant: (2)

Representative Test Data: (2)

Representation Invariant: (2)