صبا سحبان ۹۹۳۱۰۹۶

تمرین امتیازی آزمون نرم‌افزار

بخش ۱)

پیاده سازی الگوریتم ها

```
def ACoC(self) -> List[Tuple[str, ...]]:
    values = [self.characteristics[key] for key in self.characteristics]
    return list(itertools.product(*values))

def ECC(self) -> List[Tuple[str, ...]]:
    max_length = max(len(values) for values in self.characteristics.values())
    test_cases = [
        tuple(self.characteristics[key][i % len(self.characteristics[key])]
              for key in self.characteristics)
        for i in range(max_length)
    ]
    return test_cases

def BCC(self, base_choice: Dict[str, str]) -> List[Tuple[str, ...]]:
    test_cases = [tuple(base_choice[key] for key in self.characteristics)]

    for key in self.characteristics:
        original_value = base_choice[key]
        for value in self.characteristics[key]:
            if value != original_value:
                new_choice = {k: (value if k == key else base_choice[k]) for k
in self.characteristics}
                test_cases.append(tuple(new_choice[k] for k in
self.characteristics))

    return test_cases

def MBCC(self, base_tests: List[Tuple[str, ...]]) -> List[Tuple[str, ...]]:
    test_cases = {base_test for base_test in base_tests}

    for base_test in base_tests:
        base_dict = {key: base_test[i] for i, key in
enumerate(self.characteristics.keys())}
        for key in self.characteristics:
            original_value = base_dict[key]
            for value in self.characteristics[key]:
                if value != original_value:
                    new_choice = base_dict.copy()
                    new_choice[key] = value
                    test_cases.add(tuple(new_choice[k] for k in
self.characteristics))

    return list(test_cases)
```

نتایج

ACoC -1

python main.py "a=a1,a2 b=b1,b2,b3 c=c1,c2" ACoC

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "a=a1,a2 b=b1,b2,b3 c=c1,c2" ACoC


Generated test cases:
('a1', 'b1', 'c1')
('a1', 'b1', 'c2')
('a1', 'b2', 'c1')
('a1', 'b2', 'c2')
('a1', 'b3', 'c1')
('a1', 'b3', 'c2')
('a2', 'b1', 'c1')
('a2', 'b1', 'c2')
('a2', 'b2', 'c1')
('a2', 'b2', 'c2')
('a2', 'b3', 'c1')
('a2', 'b3', 'c2')
```

python main.py "x=x1,x2 y=y1,y2 z=z1,z2,z3" ACoC

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "x=x1,x2 y=y1,y2 z=z1,z2,z3" ACoC


Generated test cases:
('x1', 'y1', 'z1')
('x1', 'y1', 'z2')
('x1', 'y1', 'z3')
('x1', 'y2', 'z1')
('x1', 'y2', 'z2')
('x1', 'y2', 'z3')
('x2', 'y1', 'z1')
('x2', 'y1', 'z2')
('x2', 'y1', 'z3')
('x2', 'y2', 'z1')
('x2', 'y2', 'z2')
('x2', 'y2', 'z3')
```

python main.py "i=1,2,3 j=4,5,6 k=7,8,9" ACoC

```
Generated test cases:
('1', '4', '7')
('1', '4', '8')
('1', '4', '9')
('1', '5', '7')
('1', '5', '8')
('1', '5', '9')
('1', '6', '7')
('1', '6', '8')
('1', '6', '9')
('2', '4', '7')
('2', '4', '8')
('2', '4', '9')
('2', '5', '7')
('2', '5', '8')
('2', '5', '9')
('2', '6', '7')
('2', '6', '8')
('2', '6', '9')
('3', '4', '7')
('3', '4', '8')
('3', '4', '9')
('3', '5', '7')
('3', '5', '8')
('3', '5', '9')
('3', '6', '7')
('3', '6', '8')
('3', '6', '9')
```

ECC -2

python main.py "a=a1,a2,a3 b=b1 c=c1,c2 d=d1,d2,d3" ECC

python main.py "x=x1,x2,x3 y=y1 z=z1 z=w1,w2" ECC

python main.py "i=1,2 j=3,4,5 k=6" ECC

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "a=a1,a2,a3 b=b1 c=c1,c2 d=d1,d2,d3" ECC


Generated test cases:
('a1', 'b1', 'c1', 'd1')
('a2', 'b1', 'c2', 'd2')
('a3', 'b1', 'c1', 'd3')
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "x=x1,x2,x3 y=y1 z=z1 w=w1,w2" ECC


Generated test cases:
('x1', 'y1', 'z1', 'w1')
('x2', 'y1', 'z1', 'w2')
('x3', 'y1', 'z1', 'w1')
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "i=1,2  j=3,4,5 k=6" ECC


Generated test cases:
('1', '3', '6')
('2', '4', '6')
('1', '5', '6')
```

BCC -3

"python main.py "a=a1,a2 b=b1,b2,b3 c=c1,c2" BCC --base-choice "a=a1 b=b2 c=c1

"python main.py "x=x1,x2,x3 y=y1,y2,y3 z=z1,z2,z3" BCC --base-choice "x=x2 y=y3 z=z2

"python main.py "i=1,2,3 j=4,5,6 k=7,8" BCC --base-choice "i=2 j=5 k=8

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "a=a1,a2 b=b1,b2,b3 c=c1,c2" BCC --base-choice "a=a1 b=b2 c=c1"


Generated test cases:
('a1', 'b2', 'c1')
('a2', 'b2', 'c1')
('a1', 'b1', 'c1')
('a1', 'b3', 'c1')
('a1', 'b2', 'c2')
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "x=x1,x2,x3 y=y1,y2,y3 z=z1,z2,z3" BCC --base-choice "x=x2 y=y3 z=z2"


Generated test cases:
('x2', 'y3', 'z2')
('x1', 'y3', 'z2')
('x3', 'y3', 'z2')
('x2', 'y1', 'z2')
('x2', 'y2', 'z2')
('x2', 'y3', 'z1')
('x2', 'y3', 'z3')
```

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "i=1,2,3 j=4,5,6 k=7,8" BCC --base-choice "i=2 j=5 k=8"


Generated test cases:
('2', '5', '8')
('1', '5', '8')
('3', '5', '8')
('2', '4', '8')
('2', '6', '8')
('2', '5', '7')
(venv) sabasahban@sabas-MacBook-Pro st-bonus %
```

MBCC -4

python main.py "a=a1,a2,a3,a3,a4 b=b1,b2,b3,b4 c=c1,c2,c3,c4" MBCC --base-tests ""a=a1,b=b1,c=c1;a=a2,b=b1,c=c1

python main.py "a=a1,a2 b=b1,b2 c=c1,c2" MBCC --base-tests ""a=a1,b=b1,c=c2;a=a2,b=b2,c=c1

python main.py "x=x1,x2 b=y1,y2 c=z1,z2" MBCC --base-tests ""x=x2,b=y1,c=z1;x=x1,b=y2,c=z2

مثال جزوه:

مطابقت با جزوه:

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus %  python main.py "a=a1,a2,a3,a3,a4 b=b1,b2,b3,b4 c=c1,c2,c3,c4" MBCC --base-tests "a=a1,b=b1,c=c1;a=a2,b=b1,c=c1"


Generated test cases:
('a1', 'b1', 'c2')
('a1', 'b1', 'c4')
('a2', 'b3', 'c1')
('a1', 'b3', 'c1')
('a2', 'b1', 'c1')
('a1', 'b1', 'c1')
('a4', 'b1', 'c1')
('a2', 'b2', 'c1')
('a2', 'b1', 'c3')
('a1', 'b2', 'c1')
('a1', 'b1', 'c3')
('a2', 'b4', 'c1')
('a1', 'b4', 'c1')
('a3', 'b1', 'c1')
('a2', 'b1', 'c2')
('a2', 'b1', 'c4')
```

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "a=a1,a2 b=b1,b2 c=c1,c2" MBCC --base-tests "a=a1,b=b1,c=c2;a=a2,b=b2,c=c1"


Generated test cases:
('a2', 'b2', 'c1')
('a2', 'b2', 'c2')
('a1', 'b2', 'c1')
('a1', 'b2', 'c2')
('a2', 'b1', 'c1')
('a1', 'b1', 'c1')
('a2', 'b1', 'c2')
('a1', 'b1', 'c2')
(venv) sabasahban@sabas-MacBook-Pro st-bonus % python main.py "x=x1,x2 b=y1,y2 c=z1,z2" MBCC --base-tests "x=x2,b=y1,c=z1;x=x1,b=y2,c=z2"


Generated test cases:
('x1', 'y1', 'z2')
('x1', 'y2', 'z1')
('x1', 'y2', 'z2')
('x1', 'y1', 'z1')
('x2', 'y1', 'z1')
('x2', 'y1', 'z2')
('x2', 'y2', 'z2')
('x2', 'y2', 'z1')
```

بخش دوم:

نتایج unit test

```
(venv) sabasahban@sabas-MacBook-Pro st-bonus % coverage run -m unittest main_test.py

..........
----------------------------------------------------------------------
Ran 11 tests in 0.001s

OK
(venv) sabasahban@sabas-MacBook-Pro st-bonus % coverage report -m

Name           Stmts   Miss  Cover   Missing
--------------------------------------------
main.py           89     49    45%   52-78, 82-86, 90-94, 98-105, 109-136, 140
main_test.py      62      1    98%   101
--------------------------------------------
TOTAL            151     50    67%
```

بخش سوم:

نتایج تست bdd:

```
1 feature passed, 0 failed, 0 skipped
4 scenarios passed, 0 failed, 0 skipped
14 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.001s
```