Natnael Kebede

1001149004

1)

	1	2	3	4	5	6
t1	0	1	0	1	0	1
t2	1	0	0	0	1	0
t3	1	1	0	1	0	0
t4	1	0	1	0	0	0
t5	0	1	0	1	1	0
t6	1	0	0	0	0	0

Using procedure CMIMX to find the minimal cover set for the six entities.

Column (m) = entity to be covered and row (n) = distinct test

Step 1: minCov= φ, yetToCover = 6

Step 2: All the six tests and six entities are unmarked

Step 3: Since yetToCover > 0, we execute the loop

//Loop 1

Step 3.1: Among the unmarked entities, 3 and 6 both contain a single 1. Therefore LC = {3, 6}

Step 3.2: Among the unmarked tests, t4 covers the entities 1 and 3 while t1 covers 2, 4 and 6. Since t1 has the max number of nonzero entries, s = 1.

Step 3.3: minCov = {1}. Test t1 is marked. Additionally, entities 2, 4 and 6 covered by t1 are also marked. YetToCover = 6 – 3 = 3

//End Loop

//Loop 2: we continue with second iteration of the loop since yetToCover > 0

Step 3.1: Among the remaining entities, only 3 contains a single 1. Therfore LC = {3}

Step 3.2: Among the unmarked tests, only t4 covers the entities 1 and 3 and hence s = 4

Step 3.3: minCov = {1, 4}. Test t4 is marked. Additionally, entities 1 and 3 covered by t4 also marked. YetToCover = 3 – 2 = 1

//End Loop

//Loop 3: we continue with third iteration of the loop since yetToCover > 0

Step 3.1: Among the remaining entities, only 5 remains and contains the least number of 1s. Therefore LC = {5}

Step 3.2: Among the unmarked tests, only t2 and t5 cover the entity 5. However, both tests cover an identical benefit of 1 each in terms of the number of entities they cover. Any one of these tests (rows) would work but let's randomly select t2 and hence s = 2.

Step 3.3: minCov = {1, 4, 2}. Test t2 is marked. Additionally, entry 5 covered by t2 is also marked. YetToCover = 1– 1 = 0

The minimal cover set = {t1, t4, t2}

2)

```
----------------------------------------------------------
Test (t)      Methods covered (cov(t))         |cov(t)|
----------------------------------------------------------
t1            m1, m3, m5, m6, m8                   5
t2            m1, m7, m8                           3
t3            m1, m2, m3, m5                       4
t4            m1, m2, m3, m4                       4
t5            m1, m5, m8                           3
----------------------------------------------------------
```

Using procedure PrTest to obtain a prioritized list of tests based on residual coverage.

T': a regression test set to be prioritized
entitiesCov: set of entities covered by tests in T'
cov: Coverage vector such that for each test $t \in T'$, cov(t) is the set of entities covered by t.

Step 1: X' = {t1, t2, t3, t4, t5}. t1 covers the largest number of methods and therefore covers the largest set of entities.

Step 2: PrT = <t1>, X' = {t2, t3, t4, t5}, entitiesCov = {2, 4, 7}

Step 3: Since $X' \neq \varphi$ and $entitiesCov \neq \varphi$ we continue the steps.

Step 3.1: resCov(t) = |entitiesCov \ (cov(t) ∩ entitiesCov)|

resCov(t2) = |{2, 4, 7} \ ({1, 7, 8} ∩ {2, 4, 7})| = {2, 4} = 2

resCov(t3) = |{2, 4, 7} \ ({1, 2, 3, 5} ∩ {2, 4, 7})| = {4, 7} = 2

resCov(t4) = |{2, 4, 7} \ ({1, 2, 3, 4} ∩ {2, 4, 7})| = {7} = 1

resCov(t5) = |{2, 4, 7} \ ({1, 5, 8} ∩ {2, 4, 7})| = {2, 4, 7} = 3

Step 3.2: t4 has the least resCov or the least cost and hence is chosen.

Step 3.3: PrT = <t1, t4>, X' = {t2, t3, t5}, entitiesCov = {7}

//Since there is one more method to be covered, the loop is run one more time.

Step 3.1: resCov(t) = |entitiesCov \ (cov(t) ∩ entitiesCov)|

resCov(t2) = |{7} \ ({1, 7, 8} ∩ {7})| = |φ| = 0

resCov(t3) = |{7} \ ({1, 2, 3, 5} ∩ {7})| = {7} = 1

resCov(t5) = |{7} \ ({1, 5, 8} ∩ {7})| = {7} = 1

Step 3.2: t2 has the least resCov or the least cost and hence is chosen.

Step 3.3: PrT = <t1, t4, t2>, X' = {t3, t5}, entitiesCov = {φ}

//Since there are no more methods to be covered, we terminate the loop.

Step 4: Since the entitiesCov is empty, residual coverage criteria can't be applied to identify the priorities among the remained tests t3 and t5. Consequently, we break the tie in an arbitrary order and append them to PrT. In this case let PrT = <t1, t4, t2, t3, t5>