Homework 5

Software Engineering

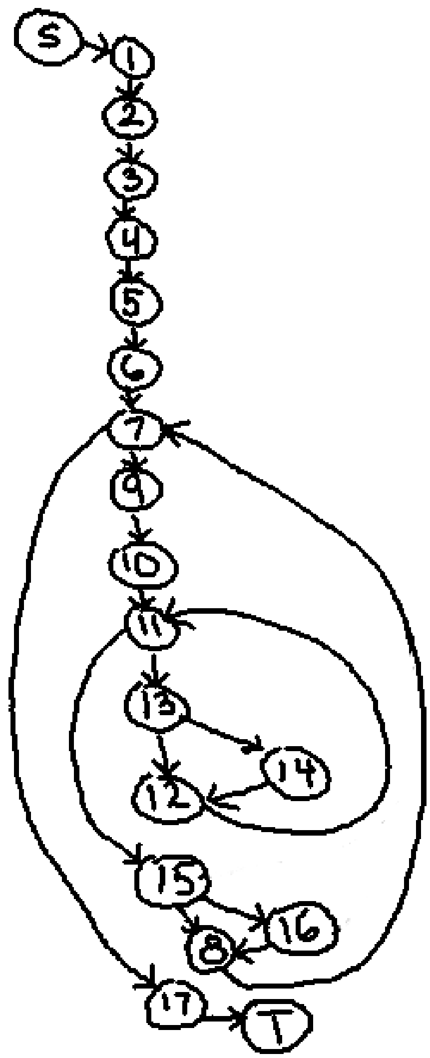
Anna Jinneman and William Roberts

Question 1

a) Flowgraph for the sieve algorithm:

Node:

```
/* Find all primes from 2-upper_bound using Sieve of Eratosthanes */

#include
typedef struct IntList {
          int value;
          struct IntList *next;
          } *INTLIST, INTCELL;
INTLIST sieve ( int upper_bound ) {

       INTLIST prime_list = NULL;   /* list of primes found */
       INTLIST cursor;              /* cursor into prime list */
       int candidate;               /* a candidate prime number */
       int is_prime;                /* flag: 1=prime, 0=not prime */

        /* try all numbers up to upper_bound */
        for (candidate=2;

             candidate <= upper_bound;
             candidate++) {

         is_prime = 1; /* assume candidate is prime */
         for(cursor = prime_list;

              cursor;
              cursor = cursor->next) {

            if (candidate % cursor->value == 0) {

              /* candidate divisible by prime */
              /* in list, can't be prime */
              is_prime = 0;
              break;  /* "for cursor" loop */
            }
         }
         if(is_prime) {

           /* add candidate to front of list */
           cursor = (INTLIST) malloc(sizeof(INTCELL));
           cursor->value = candidate;
           cursor->next  = prime_list;
           prime_list = cursor;
         }
        }
        return prime_list;
       }
```



b) 100% Node Coverage:

$T = \{t_1 = \{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 12, 11, 15, 16, 8, 7, 17\}\}$

c) 100% Edge Coverage

$T = \{t_1 = \{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 12, 11, 15, 16, 8, 7, 17\}$

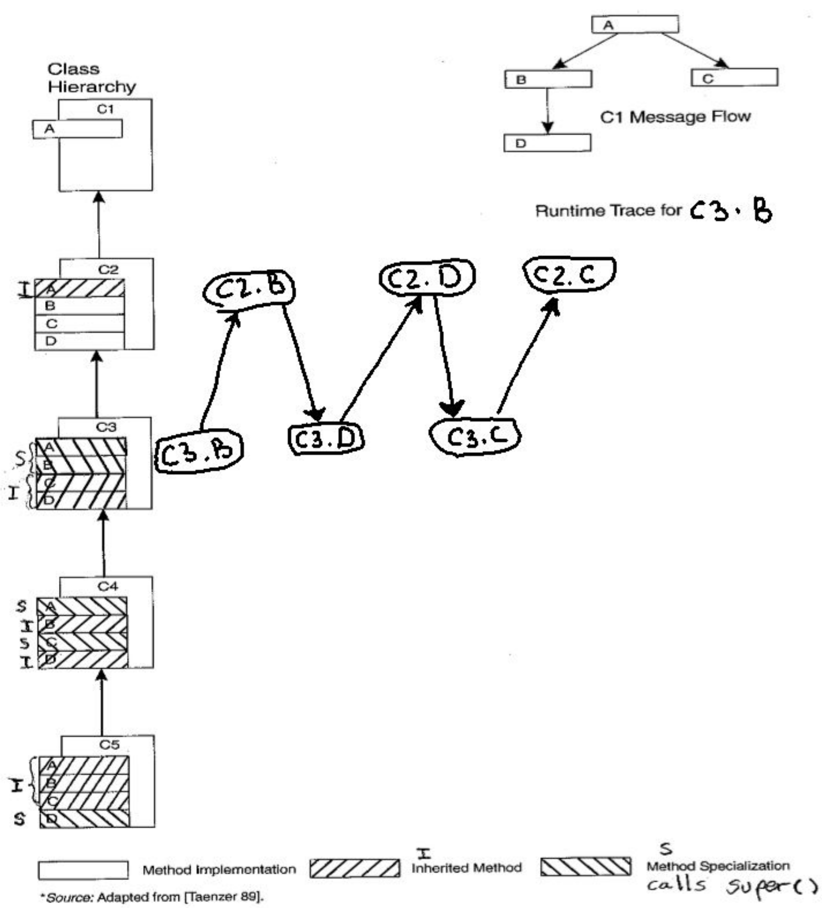
$t_2 = \{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 12, 11, 15, 8, 7, 17\}\}$

d) 100% Node or Edge cover is possible in general but for many programs it is just incredibly difficult. Many companies aim for 70-80% coverage to minimize the difficulty of testing complex programs. If it wasn't possible in general we wouldn't be able to find 100% EC and NC in these examples.

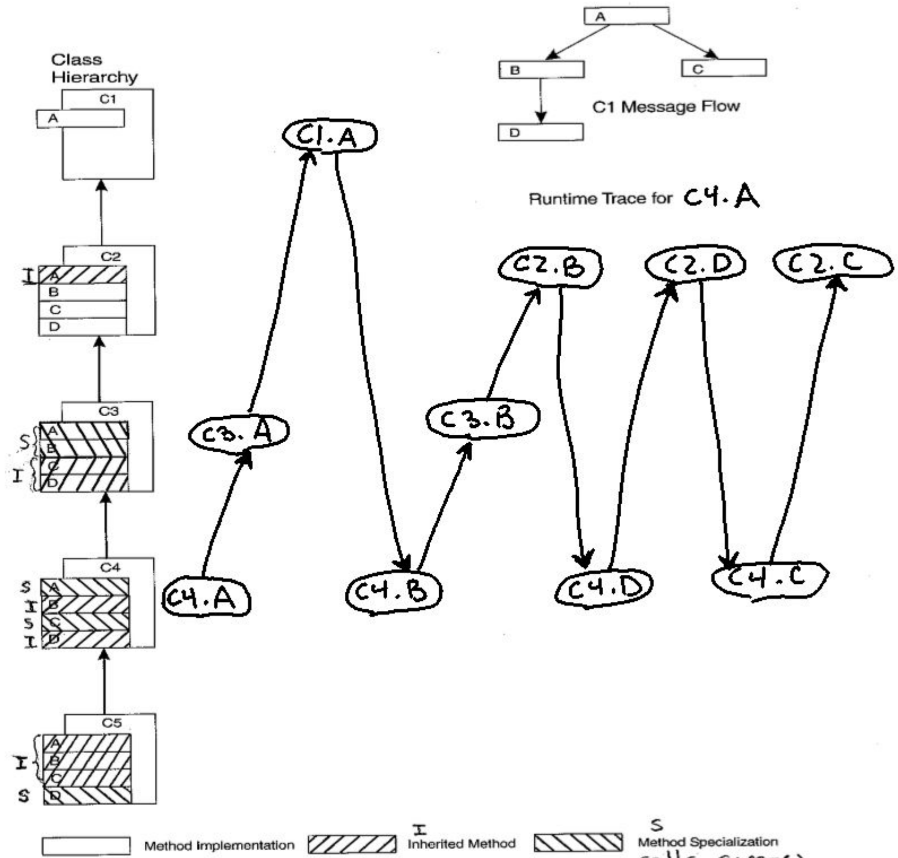
Question 2

---

a) **C3.B**



a) **C4.A**



*Source: Adapted from [Taenzer 89].

b) When C1.D is called, it will error out because there is no D method in C1 or above it in the hierarchy.

Question 3

The only input to this method is i, so each test case will only have one value. We want a test set that will break each of the mutants one by one. Listed below are the individual tests but the total test set is:
T = {t1={1}, t2={0}, t3={3}}

a) Test case to kill line 6: if (i < 1)
    t1 = {1}

b) Test case to kill line 6: if (i == 1)
    t2 = {0}

c) Test case to kill line 12: fib2 = fib;
    t3 = {3} (Any value larger than 2 works)