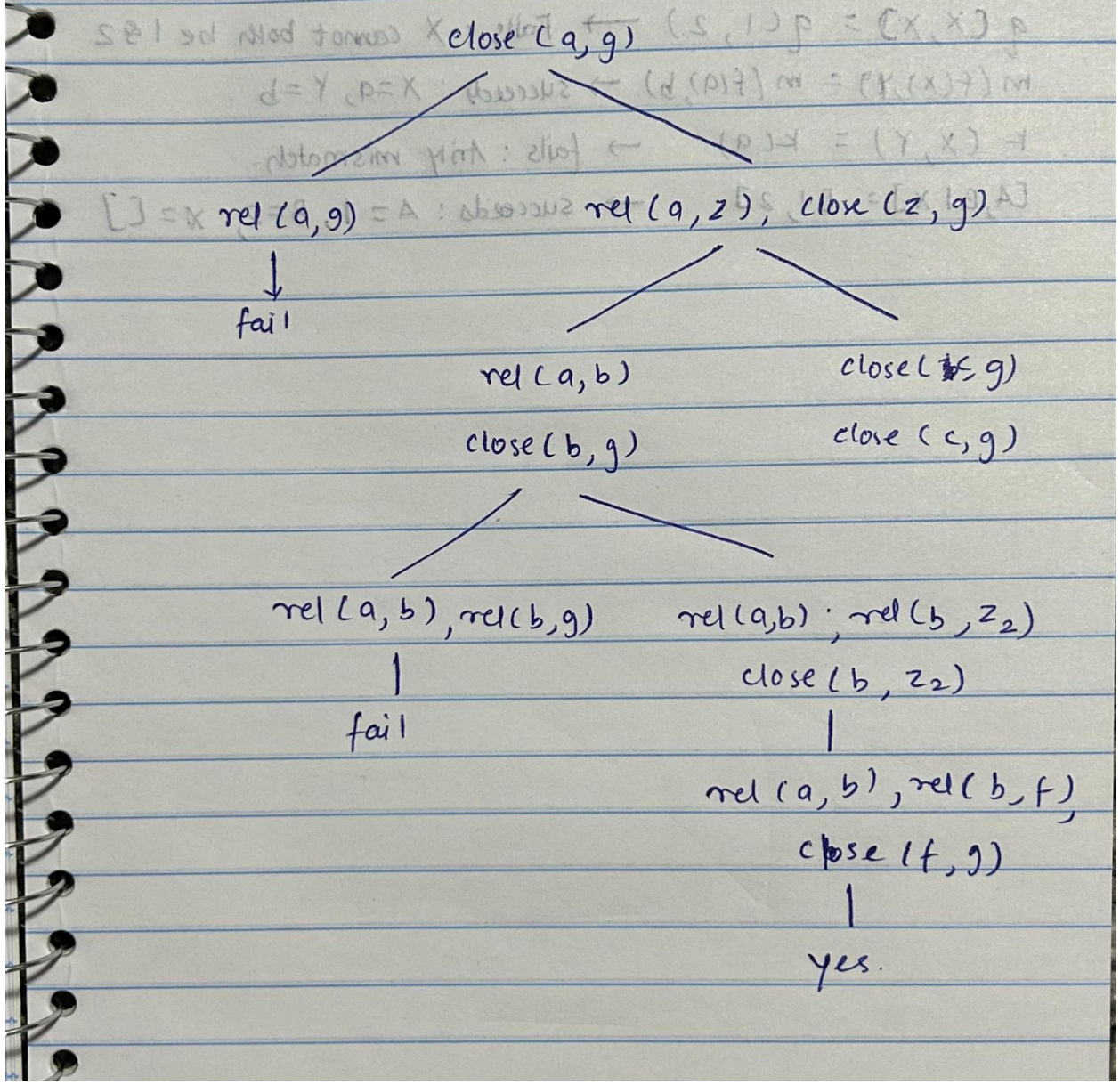
Q.1

CS571 HW-6

Q.1)

draw the search tree:



Q.2

1. p (X, Y) = p (Y , X) → Succeeds: X = Y
2. q (X, X) = q (1, 2) → Fails: X can either be 1 or 2 and not both
3. m(f(X), Y) = m(f(a), b) → Succeeds: X = a, Y = b
4. k (X, Y) = k(a) → Fails: as it has different arguments but same function name
5. [A, B | X] = [1, 2] → Succeeds: A = 1, B = 2, X = []

Q.3

In Peano arithmetic, we can formally define the predicate of exponentiation as:

```
%. Base case: any number raised to the power 0 equals 1.
exp(_, 0, s(0)).

% Recursive case: exp(X, Y , Z) if Y = s(Y1) => Z = X * exp(X, Y1)
exp(X, s(Y), Z) :-
exp(X, Y , Z1), mult(X, Z1, Z).
```

binary.pl

```
% Binary Search Tree Rotation

% rotateRight(In, Out): right rotation at root
rotateRight(tree(K2, V2, tree(K1, V1, L1, R1), R2),
          tree(K1, V1, L1, tree(K2, V2, R1, R2))).

% rotateLeft(In, Out): left rotation at root
rotateLeft(tree(K1, V1, L1, tree(K2, V2, L2, R2)),
         tree(K2, V2, tree(K1, V1, L1, L2), R2)).

% Example test tree for rotateRight
ltree(tree(3, c,
         tree(2, b,
             tree(1, a, empty, empty),
             empty),
         empty)).

% Example test tree for rotateLeft
rtree(tree(1, a,
         empty,
         tree(2, b,
             empty,
             tree(3, c, empty, empty)))).
```

Q.4

Rotate left and right predicates for a BST:

```
rotateRight(tree(Z, KZ, tree(Y , KY , A, B), C), tree(Y , KY , A, tree(Z, KZ, B, C))).
rotateLeft(tree(X, KX, A, tree(Y , KY , B, C)), tree(Y , KY , tree(X, KX, A, B), C)).
ltree(tree(3, c, tree(2, b, tree(1, a, empty, empty), empty), empty)).
rtree(tree(1, a, empty, tree(2, b, empty, tree(3, c, empty, empty)))).
```

peano.pl

```
is_number(0).
is_number(s(X)) :- is_number(X).

add(0, N, N) :- is_number(N).
add(s(N), M, s(Y)) :- add(N, M, Y).

mul(0, N, 0) :- is_number(N).
mul(s(N), M, Y) :-
  mul(N, M, YY),
  add(M, YY, Y).

% exp(X, Y, Z) is true when X^Y = Z, with X, Y, Z in Peano representation.
% Base case: any number to the 0th power is 1 (s(0)).
exp(X, 0, s(0)) :-
  is_number(X).

% Recursive case: X^(s(Y)) = X * (X^Y)
exp(X, s(Y), Z) :-
  is_number(X),
  exp(X, Y, Z1),
  mul(X, Z1, Z).

% Example queries:
% ?- exp(s(s(0)), s(s(0)), Z).     % 2^2 = 4
% Z = s(s(s(s(0)))).
% ?- exp(s(s(0)), s(s(s(0))), Z). % 2^3 = 8
% Z = s(s(s(s(s(s(s(s(0)))))))).
```