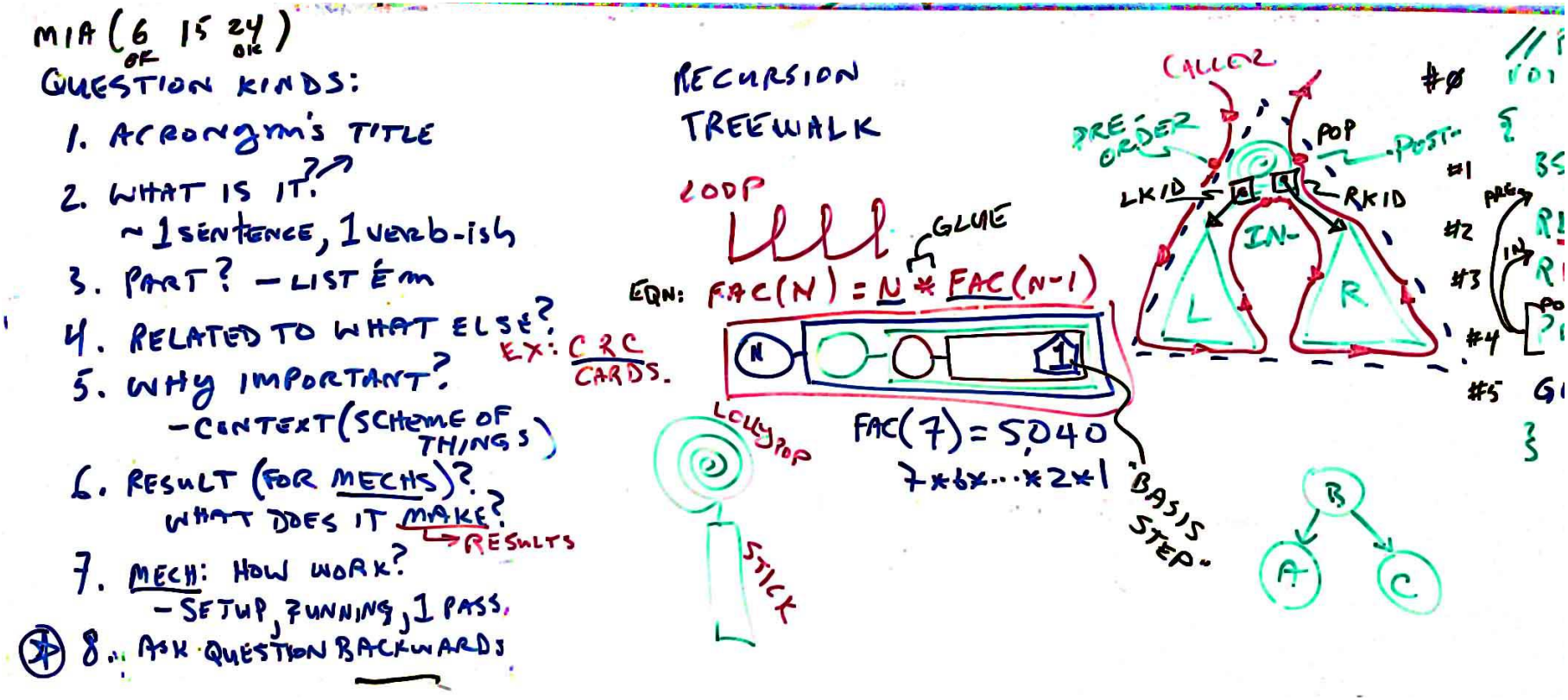
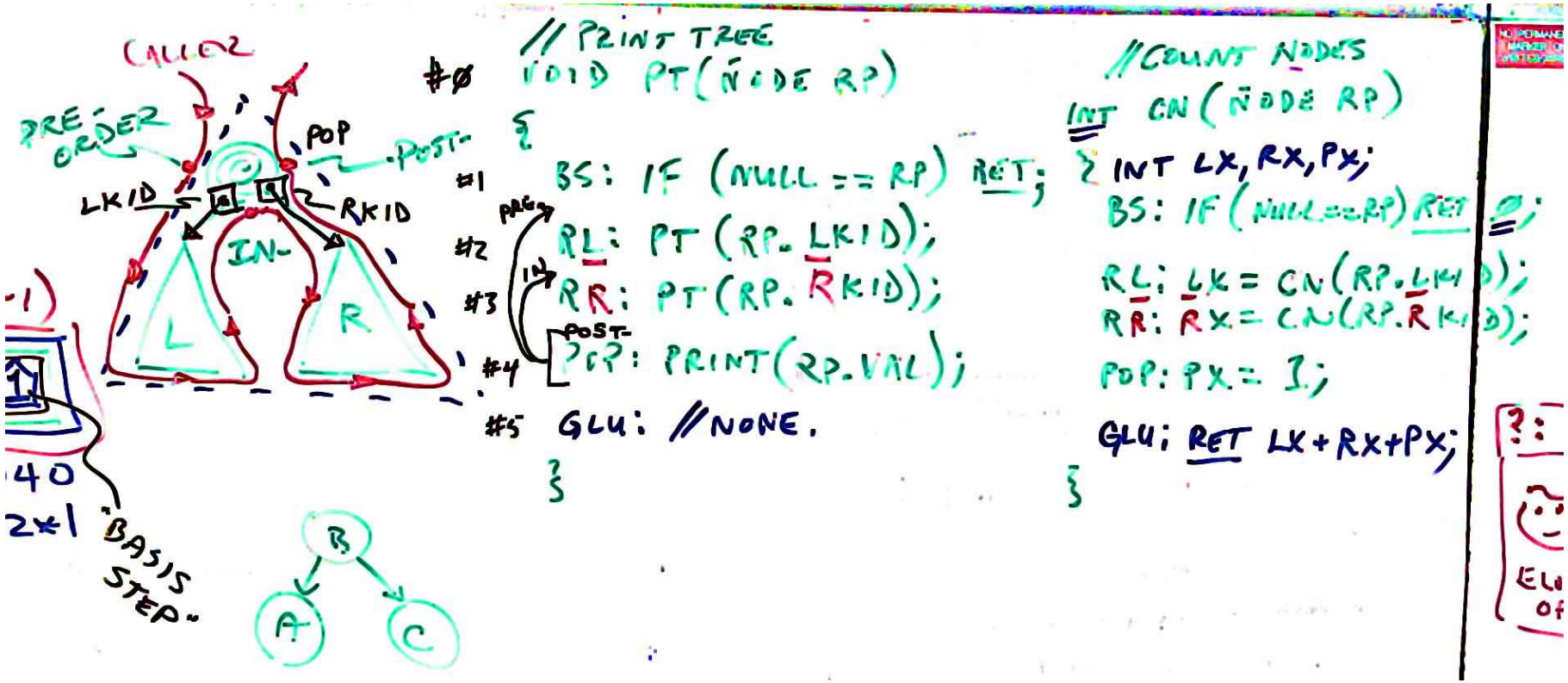
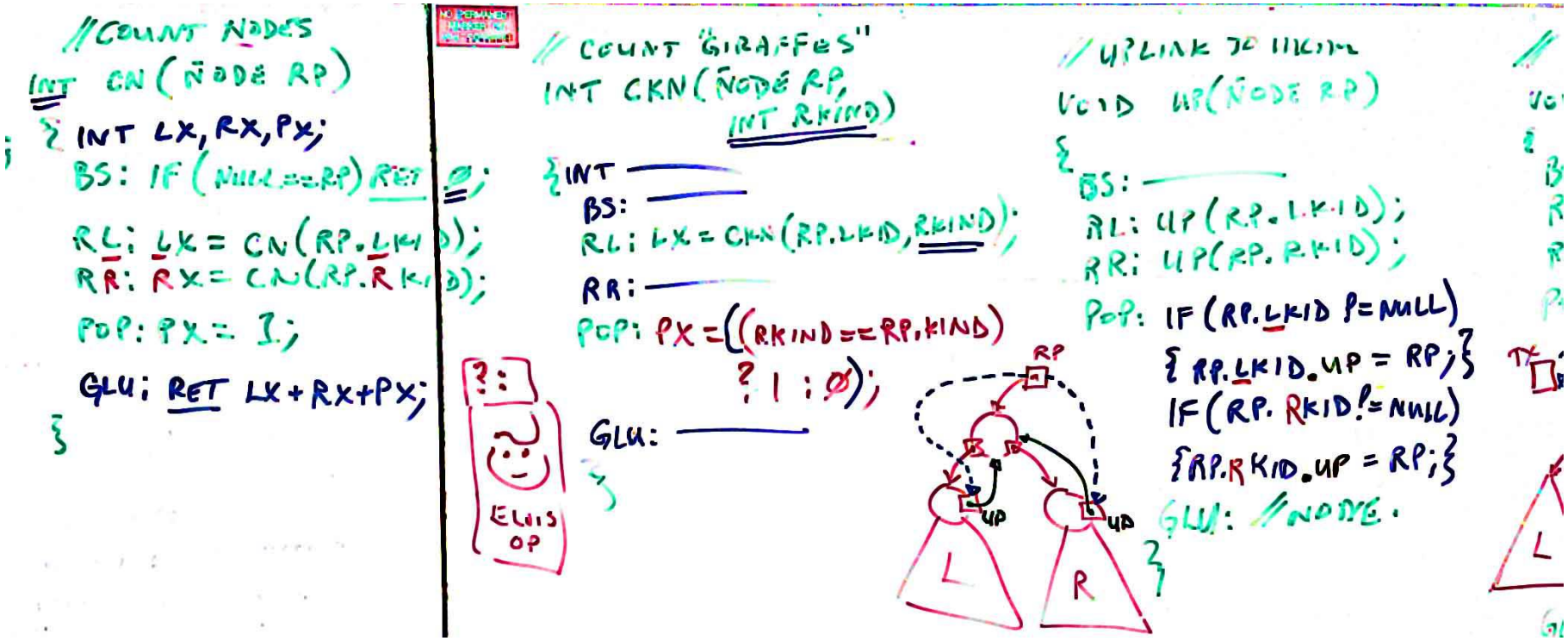
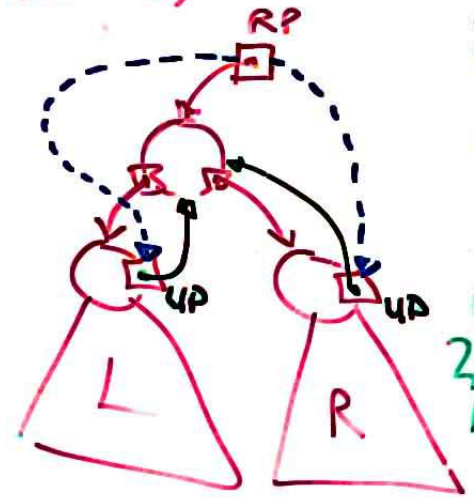

MIA (6 OR 15 24 OK)
QUESTION KINDS:
1. ACRONYM'S TITLE
2. WHAT IS IT?
~1 SENTENCE, 1 VERB-ish
3. PART? - LIST 'EM
4. RELATED TO WHAT ELSE?
EX: CRC CARDS.
5. WHY IMPORTANT?
-CONTEXT (SCHEME OF THINGS)
6. RESULT (FOR MECHS)?
WHAT DOES IT MAKE?
RESULTS
7. MECH: HOW WORK?
-SETUP, RUNNING, 1 PASS.
8. ASK QUESTION BACKWARDS
RECURSION
TREEWALK
LOOP
GLUE
EQN: FAC(N) = N * FAC(N-1)
N
1
LOLLYPOP
STICK
FAC(7) = 5,040
7*6*...*2*1
BASIS STEP
CALLER
PRE ORDER
POP
POST
LKID
RKID
IN-
L
R
B
A
C


CALLER
PRE-ORDER
POP
POST-
LKID
RKID
IN-
L
R
-1)
40
2*1
BASIS
STEP.
B
A
C
// PRINT TREE
#0 VOID PT(NODE RP)
{
#1 BS: IF (NULL == RP) RET;
#2 RL: PT (RP. LKID);
#3 RR: PT(RP. RKID);
#4 PO?: PRINT(RP.VAL);
#5 GLU: //NONE.
}
PRE
IN
POST-
// COUNT NODES
INT CN (NODE RP)
{ INT LX, RX, PX;
BS: IF (NULL==RP) RET
RL; LX = CN(RP.LKI
RR: RX = CN(RP.RKI
POP: PX = 1;
GLU; RET LX+RX+PX;
}


// COUNT NODES
INT CN(NODE RP)
{ INT LX, RX, PX;
BS: IF (NULL==RP) RET 0;
RL: LX = CN(RP.LKID);
RR: RX = CN(RP.RKID);
POP: PX = 1;
GLU: RET LX+RX+PX;
}
// COUNT "GIRAFFES"
INT CKN(NODE RP, INT RKIND)
{INT
BS:
RL: LX = CKN(RP.LKID, RKIND);
RR:
POP: PX = ((RKIND==RP.KIND) ? 1 : 0);
GLU:
}
ELSE OP
// UPLINK TO PARENT
VOID UP(NODE RP)
{
BS:
RL: UP(RP.LKID);
RR: UP(RP.RKID);
POP: IF (RP.LKID != NULL)
{RP.LKID.UP = RP;}
IF (RP.RKID != NULL)
{RP.RKID.UP = RP;}
GLU: // NONE.
}
RP
UP
UP
L
R

```
// UPLINK TO HIGHER
VOID UP(NODE RP)
{
  BS: ————————
  RL: UP(RP.LKID);
  RR: UP(RP.RKID);
  POP: IF (RP.LKID != NULL)
       { RP.LKID.UP = RP; }
       IF (RP.RKID != NULL)
       { RP.RKID.UP = RP; }
  GLU: //NODE.
}
```




```
// SWAP KIDS
VOID SWP(NODE RP)
{
  BS: ————————
  RL: ——— SWP ———
  RR: ————————"————————
  POP: { TX = RP.LKID;
         RP.LKID = RP.RKID;
         RP.RKID = TX;
  GLU: //NONE.
}
```