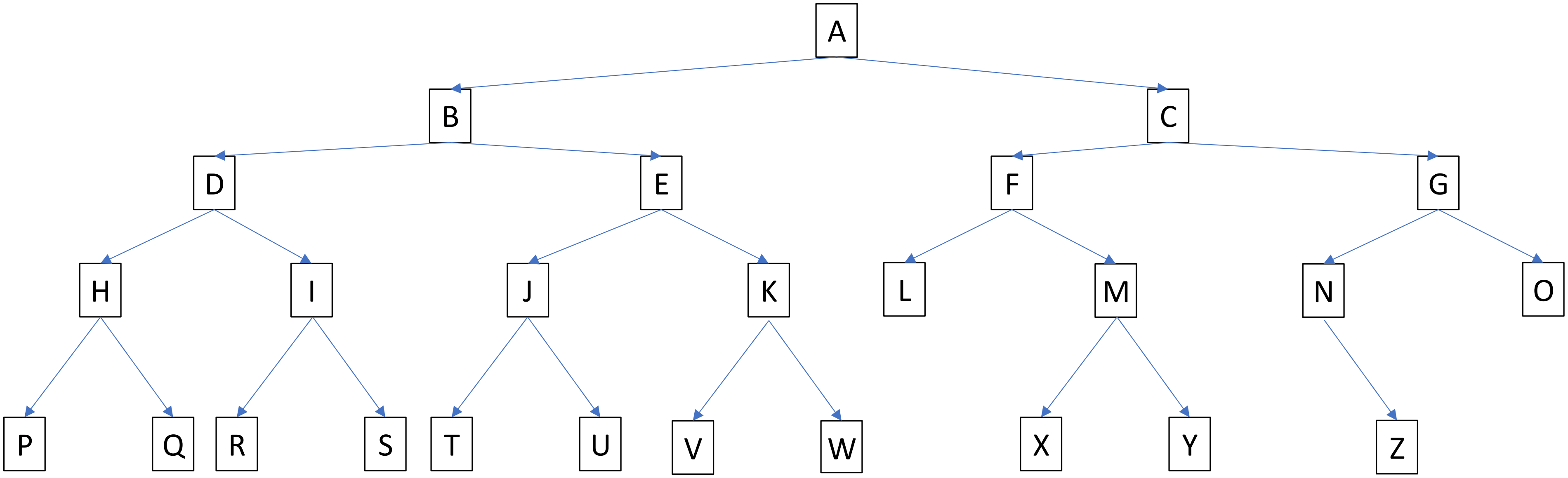
Homework

READ Chapters 6 and 7 of “grokking algorithms” – try all exercises, but do NOT turn in! (yes we are skipping chapter 5 for now)

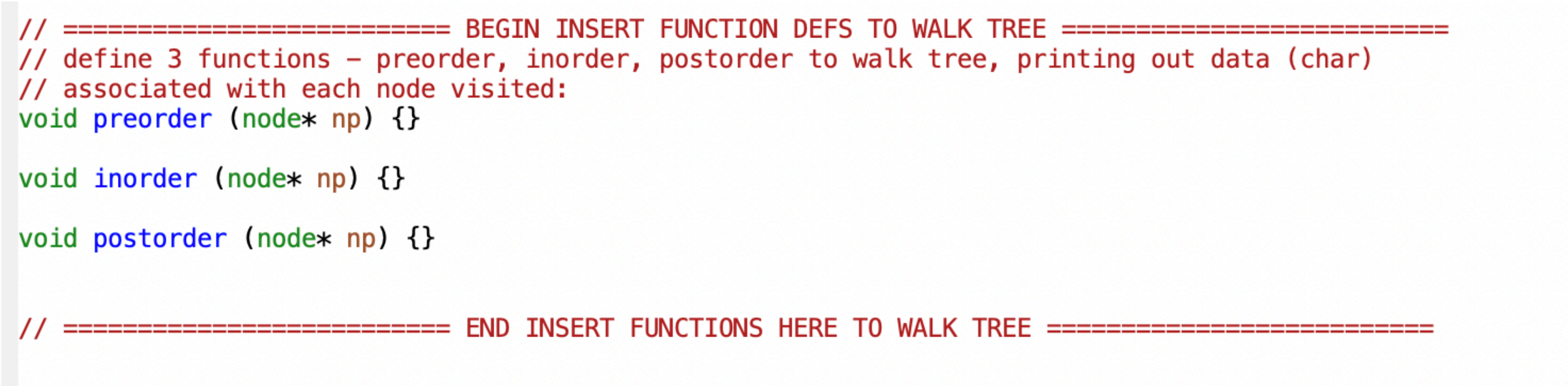
WATCH the videos on topological sorting

Homework

#1 Program a preorder, inorder, postorder traversal of this tree – print the node values “processed” in each. Modify `walk.c` in github – it creates the tree for you. All you have to do is fill in 3 functions.



Homework – place to insert code in `walk.c`



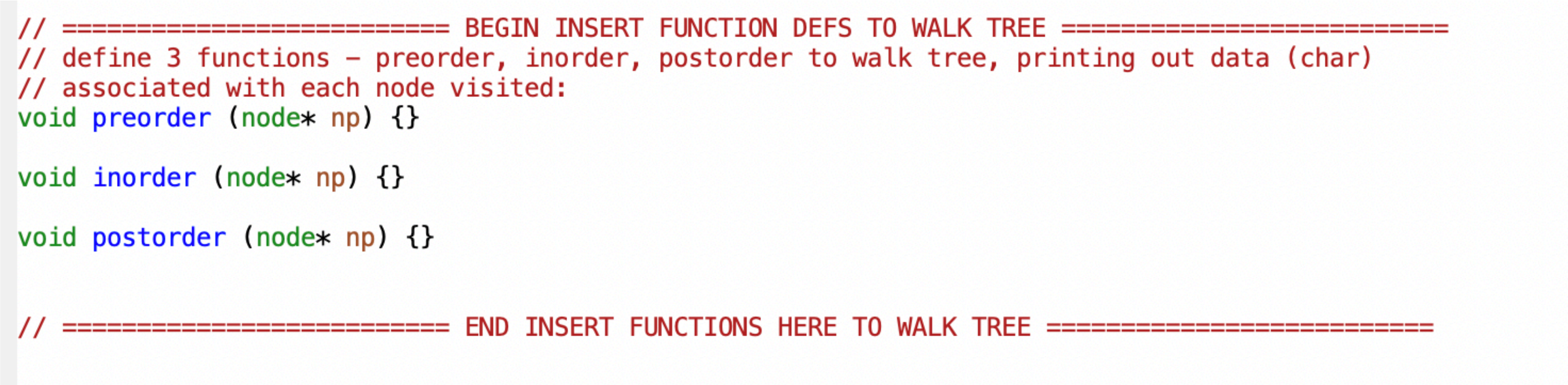
```
// ========================== BEGIN INSERT FUNCTION DEFS TO WALK TREE ==========================
// define 3 functions - preorder, inorder, postorder to walk tree, printing out data (char)
// associated with each node visited:
void preorder (node* np) {}

void inorder (node* np) {}

void postorder (node* np) {}


// ========================== END INSERT FUNCTIONS HERE TO WALK TREE ==========================
```

Homework – place to insert code in `walk.c`

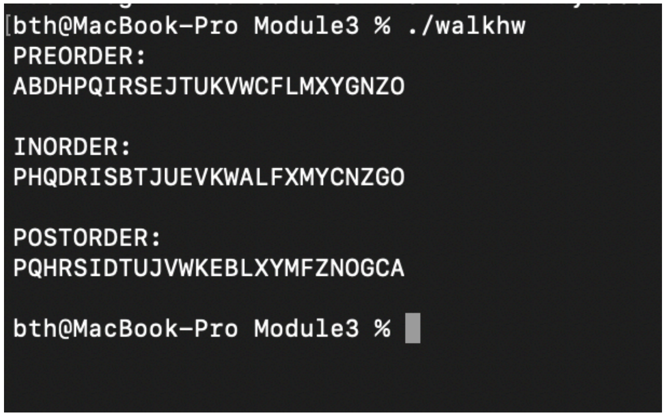


```
// ======================== BEGIN INSERT FUNCTION DEFS TO WALK TREE ========================
// define 3 functions - preorder, inorder, postorder to walk tree, printing out data (char)
// associated with each node visited:
void preorder (node* np) {}

void inorder (node* np) {}

void postorder (node* np) {}


// ======================== END INSERT FUNCTIONS HERE TO WALK TREE ========================
```



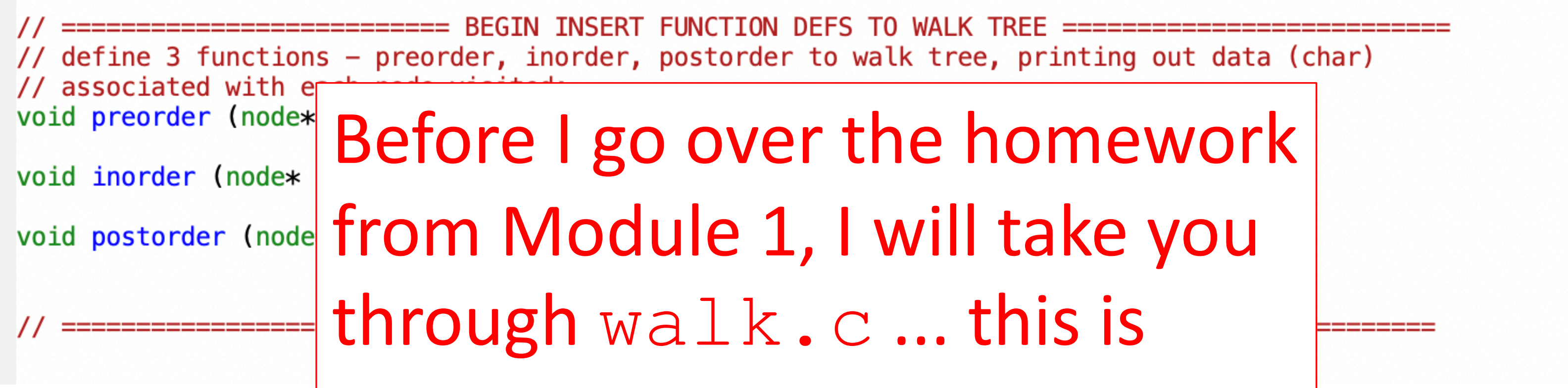
```
[bth@MacBook-Pro Module3 % ./walkhw
PREORDER:
ABDHPQIRSEJTUKVWCFLMXYGNZO

INORDER:
PHQDRISBTJUEVKWALFXMYCNZGO

POSTORDER:
PQHRSIDTUJVWKEBLXYMFZNOGCA

bth@MacBook-Pro Module3 %
```

Homework – place to insert code in `walk.c`



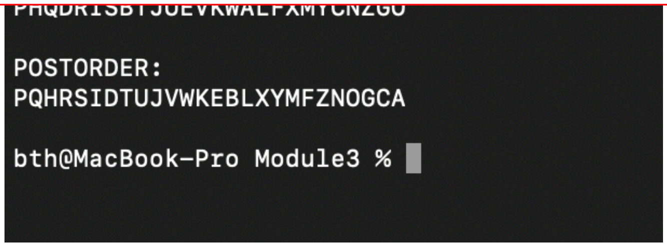
```
// ========================= BEGIN INSERT FUNCTION DEFS TO WALK TREE =========================
// define 3 functions - preorder, inorder, postorder to walk tree, printing out data (char)
// associated with e
void preorder (node*

void inorder (node*

void postorder (node


// ================                                                                  ========
```

Before I go over the homework from Module 1, I will take you through `walk.c` ... this is optional, so feel free to try to figure the code out yourself.



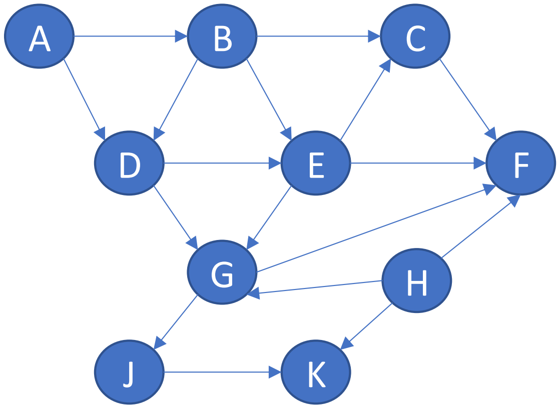
```
PHQDRISBTJUEVKWALFXMYCNZGO

POSTORDER:
PQHRSIDTUJVWKEBLXYMFZNOGCA

bth@MacBook-Pro Module3 %
```

Homework – page 4 of 4

#2 For this graph write down the order of vertices encountered in a breadth-first search starting from vertex A. Break ties by picking the vertices in alphabetical order (for example, A before Z)



Just turn in a pdf with the results.

Homework – due 6pm PT next Tuesday

Please upload to Canvas your C program and screen shot of you executing the program.

For this homework assignment, you may (if you choose) work in teams of 2.

Each student is expected to work WITH the other student while writing and testing (via ZOOM or whatever collaboration tool you like). Do not split the task between the two of you.

Each student should upload the program/screen shot + pdf of the graph traversal results

The comment at the top of the program should say if it was done by a team and who the two members of the team were (so that the TAs do not have to grade the same homework twice).