Final Quiz Review

CS 5008 Spring 2023

Final Quiz: Purpose and General Topics

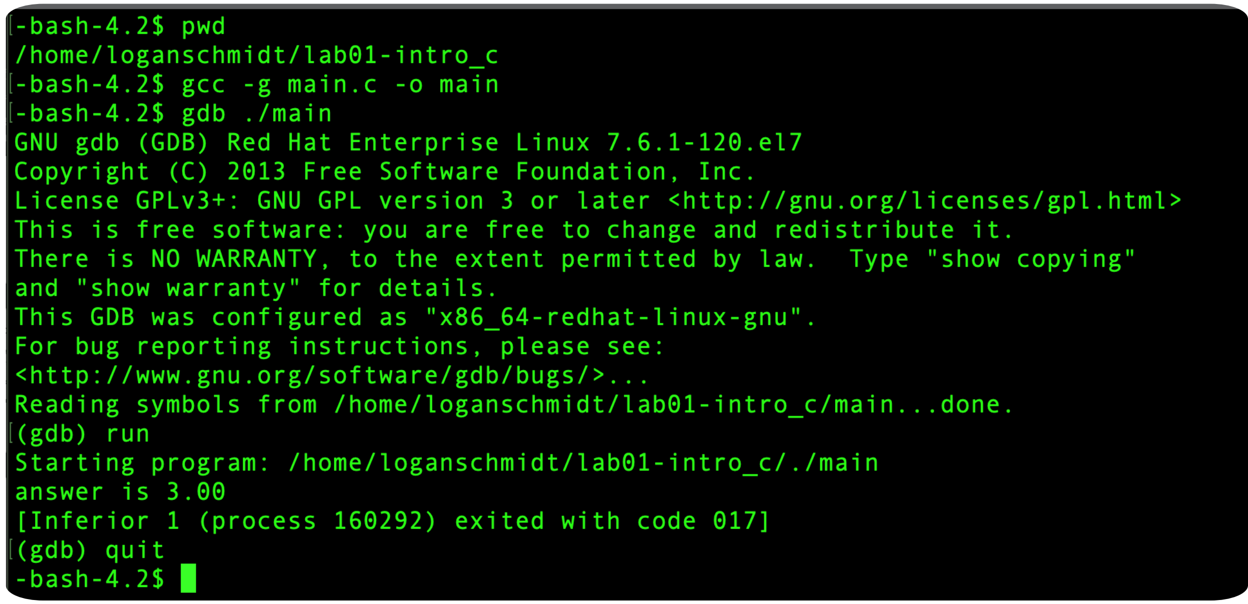
- The final quiz is meant as a culmination of your learning this semester
- It is meant to cover the fundamentals of what we've learned in this class
- It is comprehensive and can cover any topics that we've covered during this course
- It does not include any material covered by student presentations or by the final project

Content Covered on this Final Quiz

- Debugging in C and debugging environment
- Solving a recurrence equation
- Describing features of sorts and sorting algorithms
- Basic assembly instructions
- Trees, Binary search trees, operations
- Big O of basic operations on data structures
- Basics of computer networks
- Dijkstra’s algorithm
- Graph traversals
- Graph representations
- Greedy algorithms
- Dynamic Programming

Debugging in C: Advice for review

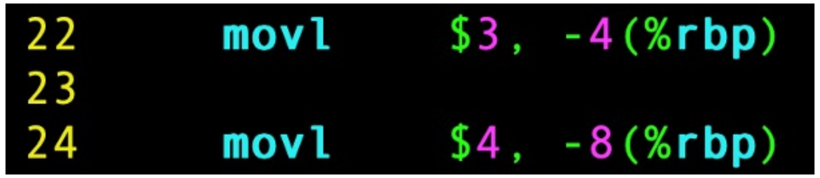
- Run GDB at least once on a simple program (HW1, Lab 1, etc.)
- If you haven't, review basics at
  - https://northeastern.instructure.com/courses/103095/pages/debugging-with-gdb-gui-25-20?module_item_id=7038137



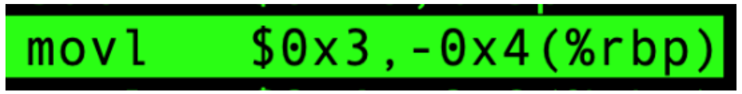
```
-bash-4.2$ pwd
/home/loganschmidt/lab01-intro_c
-bash-4.2$ gcc -g main.c -o main
-bash-4.2$ gdb ./main
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-120.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/loganschmidt/lab01-intro_c/main...done.
(gdb) run
Starting program: /home/loganschmidt/lab01-intro_c/./main
answer is 3.00
[Inferior 1 (process 160292) exited with code 017]
(gdb) quit
-bash-4.2$ 
```

Basic assembly instructions

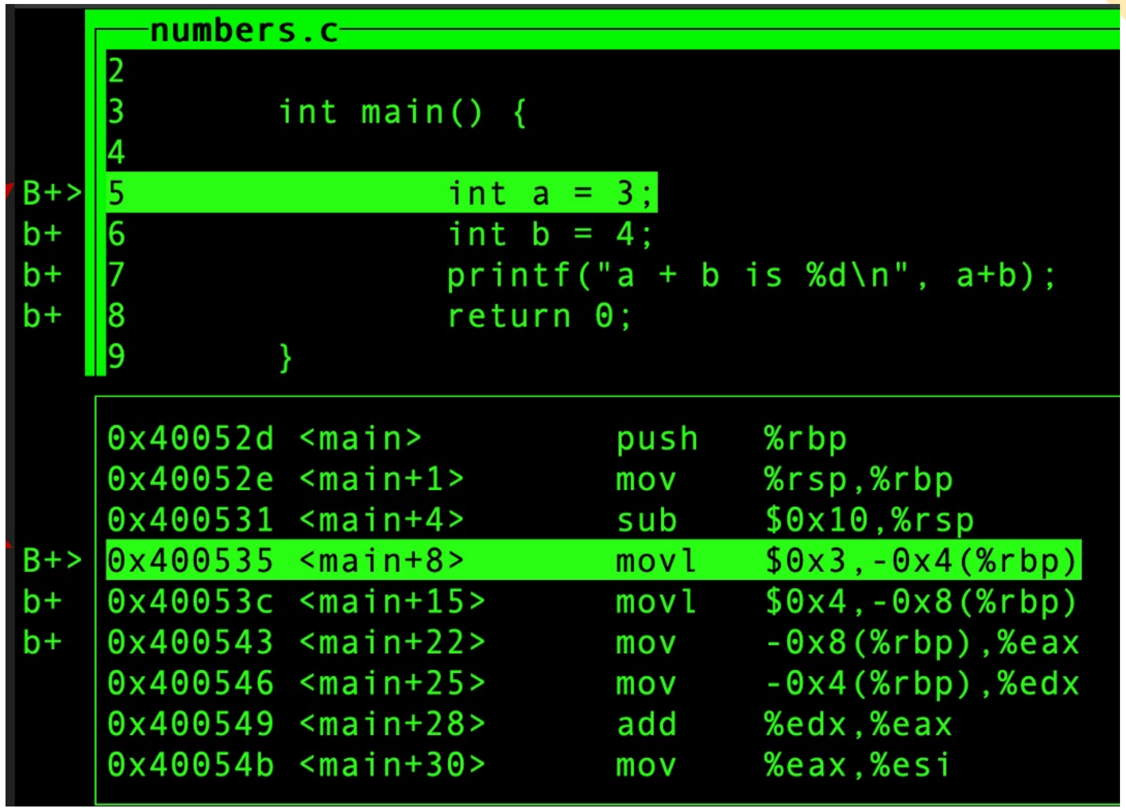
- Be able to read a basic assembly instruction
- Source, destination or value, destination
- Understand memory address arithmetic



```
22        movl      $3,  -4(%rbp)
23
24        movl      $4,  -8(%rbp)
```



```
movl    $0x3,-0x4(%rbp)
```



```
 numbers.c
2
3        int main() {
4
B+> 5            int a = 3;
b+  6            int b = 4;
b+  7            printf("a + b is %d\n", a+b);
b+  8            return 0;
9        }

    0x40052d <main>         push   %rbp
    0x40052e <main+1>       mov    %rsp,%rbp
    0x400531 <main+4>       sub    $0x10,%rsp
B+> 0x400535 <main+8>       movl   $0x3,-0x4(%rbp)
b+  0x40053c <main+15>      movl   $0x4,-0x8(%rbp)
b+  0x400543 <main+22>      mov    -0x8(%rbp),%eax
    0x400546 <main+25>      mov    -0x4(%rbp),%edx
    0x400549 <main+28>      add    %edx,%eax
    0x40054b <main+30>      mov    %eax,%esi
```

Be able to solve a recurrence equation using any method

Master’s theorem

Substitution method

Recurrence tree

Review what they are, how they work

Be able to solve a simple recurrence equation

Basic operations and big O of basic operations on data structures

- Singly linked list
- Doubly linked list
- Queue
- Stack
- BST
- Heap

Describing features of sorts

(and binary search)

- Big O of space and time
- Best case, worst case data
- Stable vs unstable
- Quadratic sorts, divide and conquer algorithms for sorting

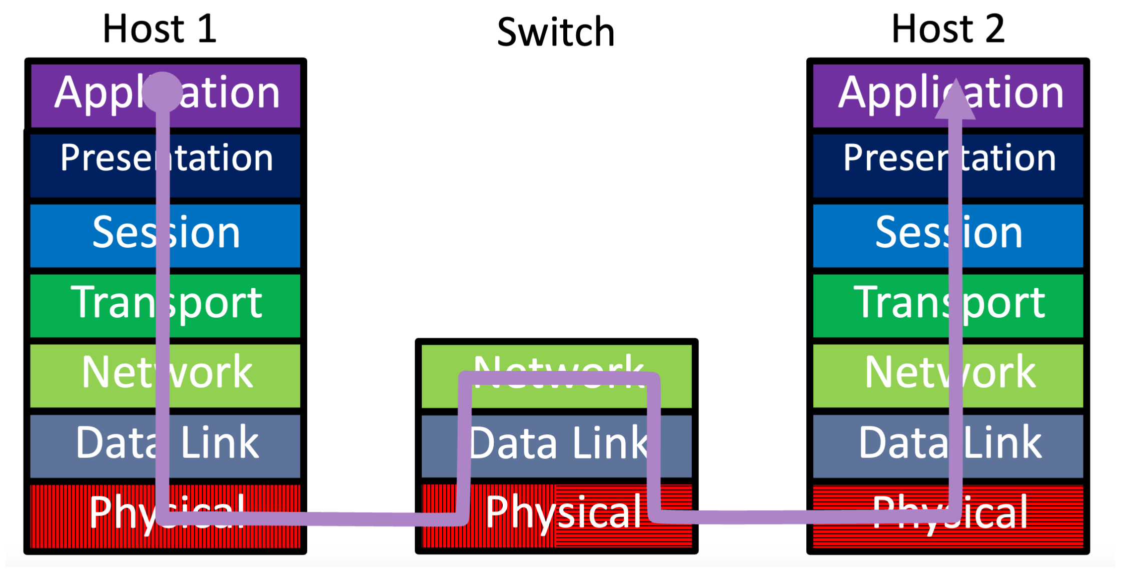
- Of course understand linear & binary search

Basics of computer networks

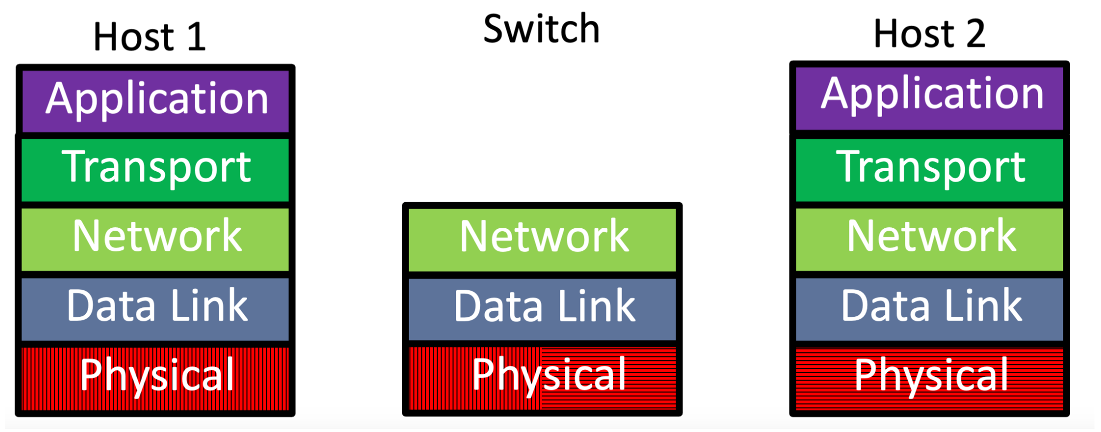
- How does a packet move through the OSI layers on the internet?
- What are the 5 most important layers of the OSI layer model?
- What are the differences between TCP and UDP, and why/when would you use one rather than the other?
- What are the major differences between IPv4 and IPv6 addresses?

2001:0db8::ff00:42:8329
vs
192.163.12.1

How packets travel on the internet

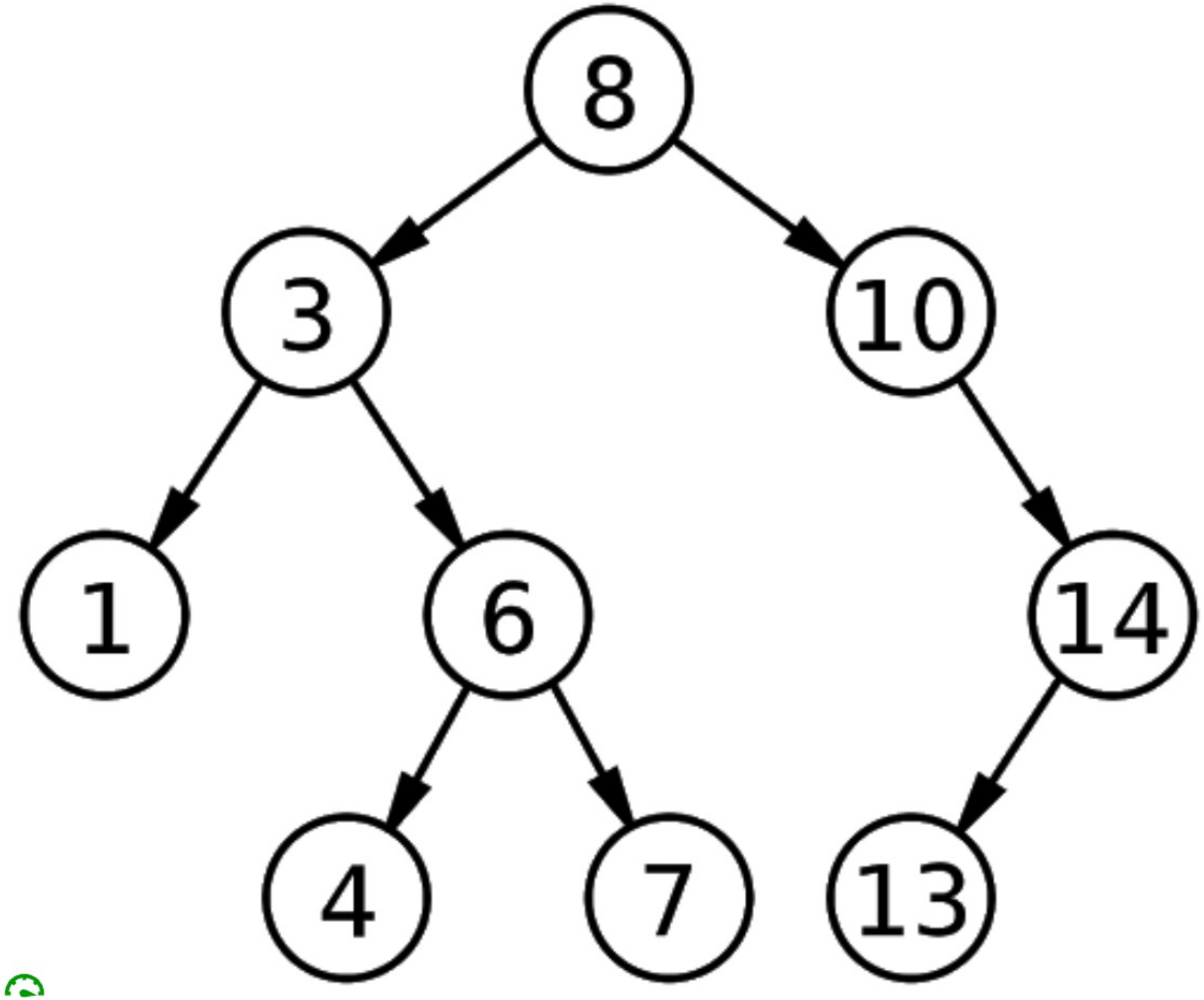


The network stack in practice



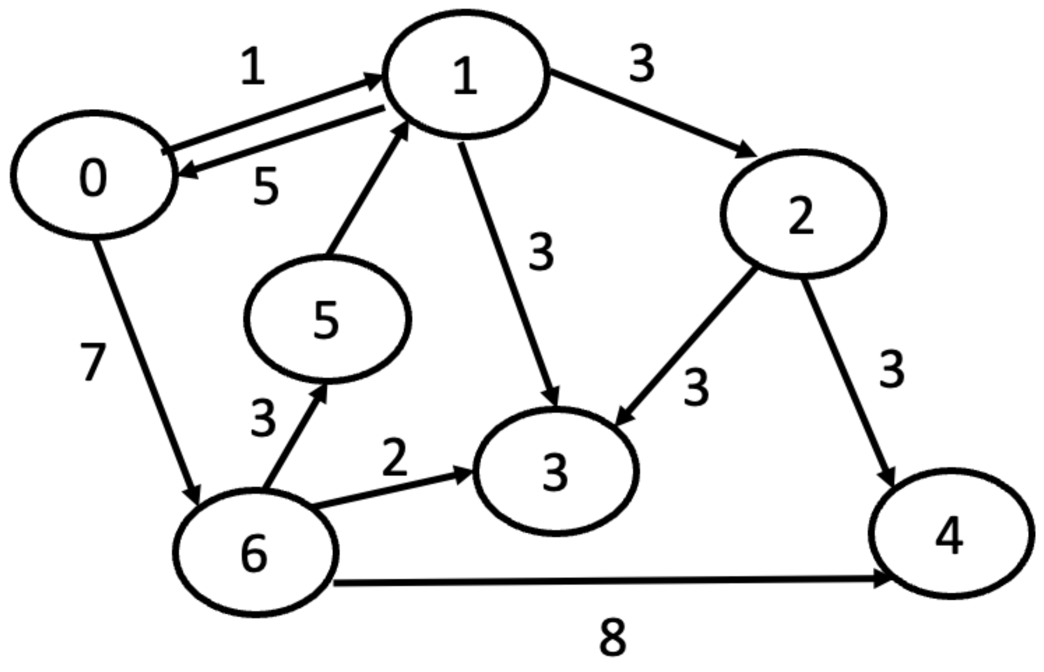
Binary search tree operations

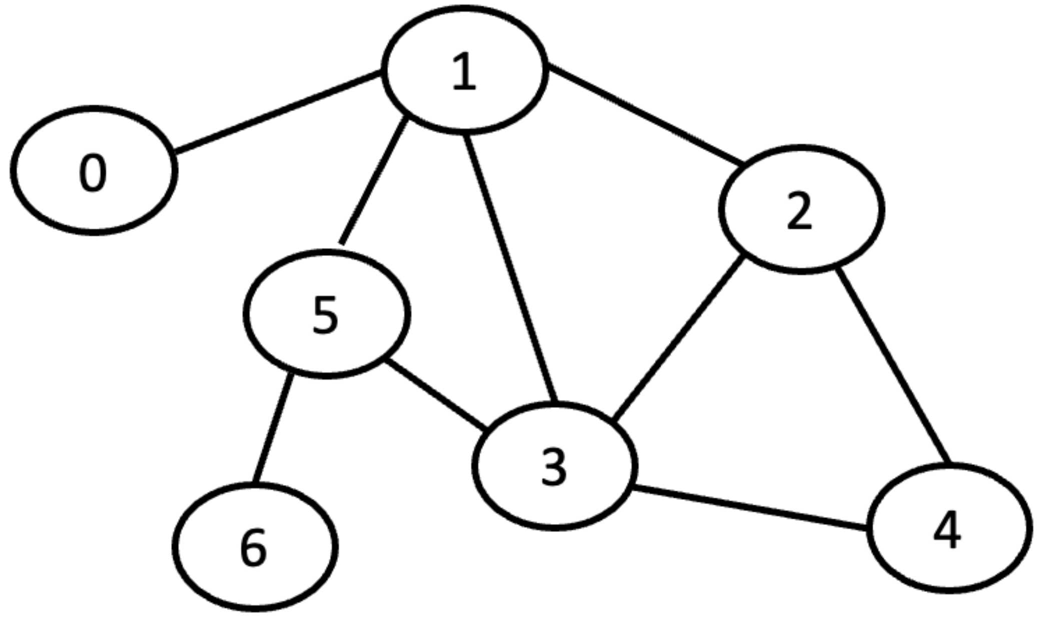
- add, delete, search
- Big O of operations
- Be able to perform these operations on a BST



Graph representations

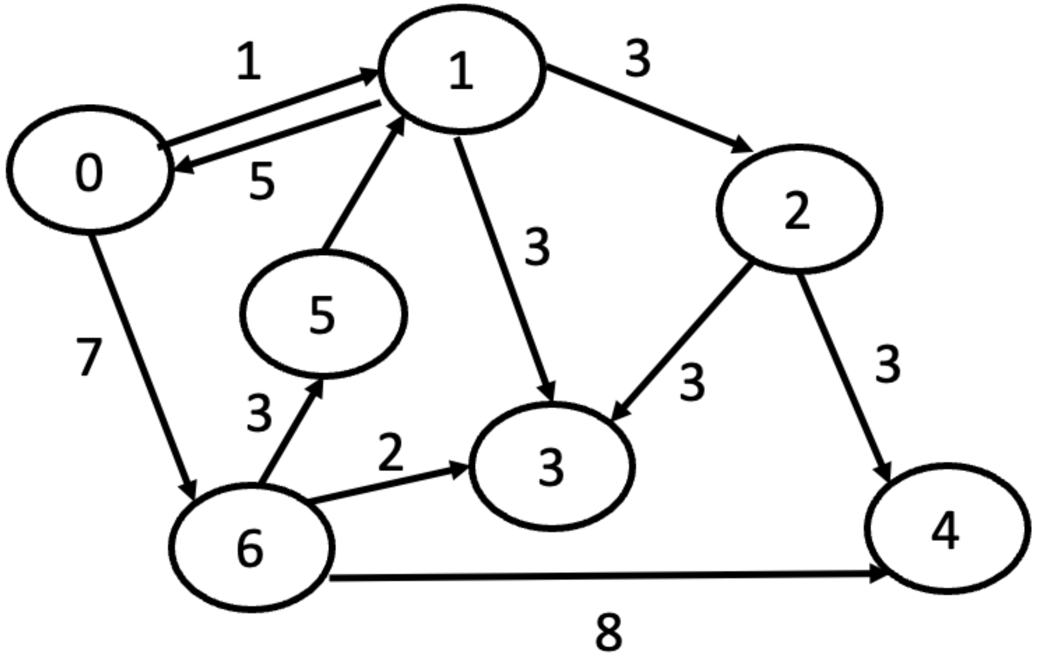
- Adjacency matrix – read and create/draw
- Adjacency list – read and create/draw

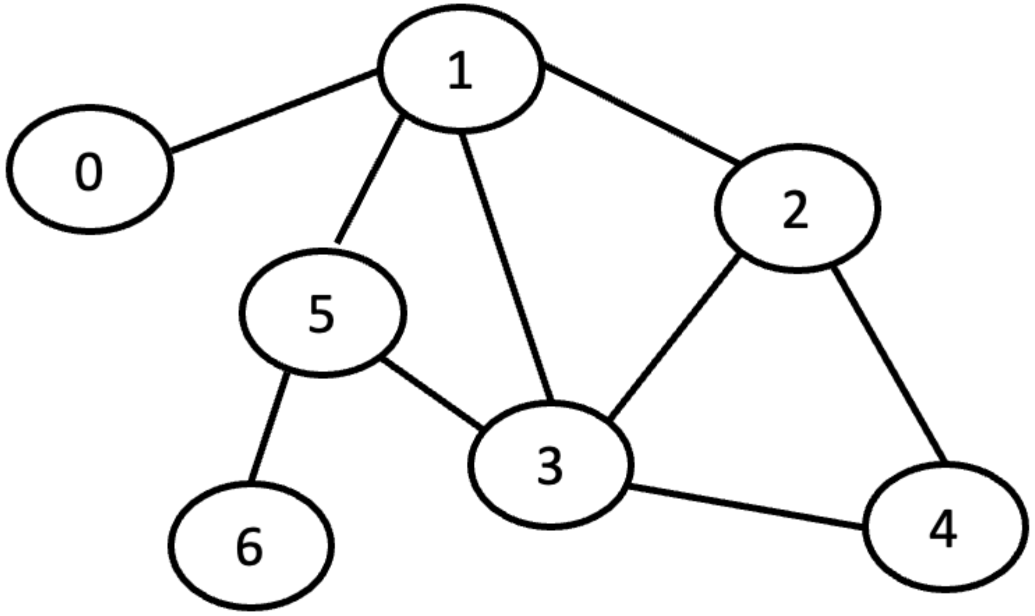




Graph traversals

- Breadth-first search – be able to perform/draw
- Depth-first search – be able to perform/draw



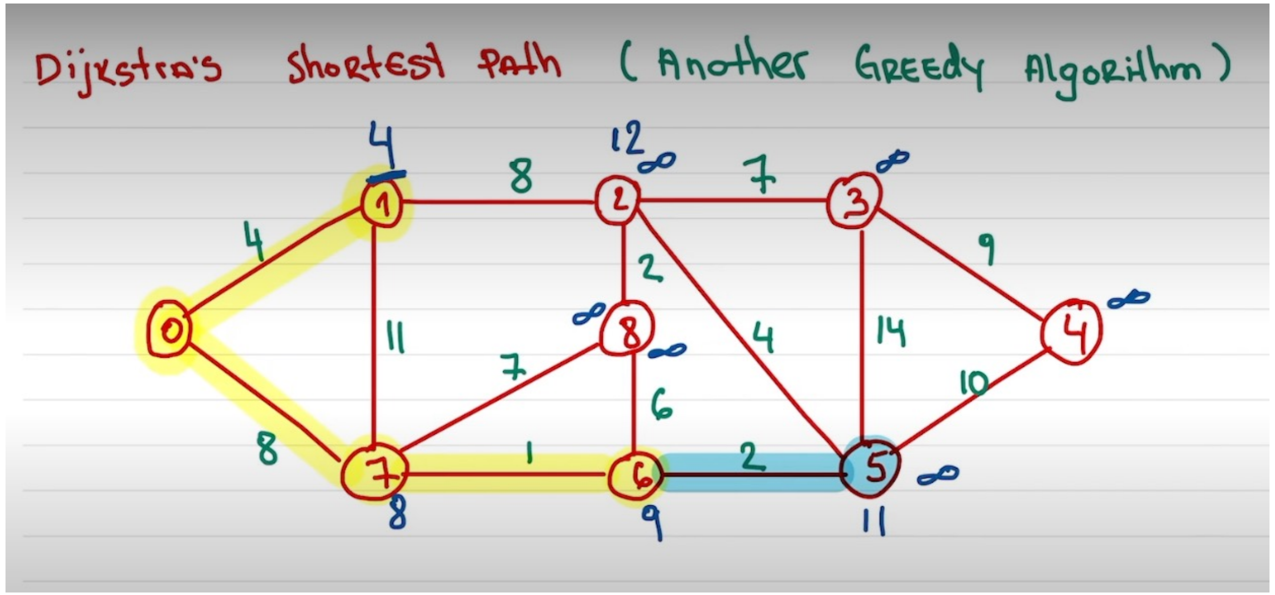


Greedy algorithms (including Dijkstra’s)

- What do they have in common?
- How do greedy solutions in general work?
- How do you know if a greedy solution is also a correct solution?

Dijkstra's algorithm

- What problem does it solve
- What kind of data structure(s) does it require, and how does it interact/use those data structures
- How does it work
- Be able to perform Dijkstra's on graph



Dynamic Programming

- What do dynamic programming solutions have in common?
- What is memoization and how does it work?
- How do DP solutions in general work?

Congratulations!

Thank you for a great semester!

TRACE