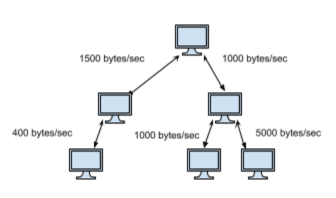
Please use this Google doc during your interview (your interviewer will see what you write here). To free your hands for typing, we recommend using a headset or speakerphone.

There is a lot of static on my end, can you type the question? Thanks :)

I thought you had a question for me, sorry, no worries if you don’t.



Given a root node, and a filesize of N bytes, calculate the time to transmit the file from the root node to the whole network.

class Edge:

def \_\_init\_\_(self, node, rate):

self.node = node

self.rate = rate

class Node:

def \_\_init\_\_(self, id):

self.id = id

self.edges = []

def add(self, new\_id, rate):

new\_edge = Edge(Node(new\_id), rate)

self.edges.append(new\_edge)

return new\_edge.node

root = Node(‘a’)

b = root.add(‘b’,1500)

c = root.add(‘c’,1000) etc.

# 10000.0

def transmit\_time(node, file\_size):

stack\_computer = []

stack\_receive = [] # receiving rate of current computer

stack\_time = []

visited = set() # the id of the computers visited

max\_time = 0

# if there’s not computer

if node is None:

return 0

stack\_receive.push(float(‘inf’))

stack\_computer.push(node)

stack\_time.push(0)

while stack\_computer:

cur\_receive\_rate = stack\_receive.pop()

cur\_computer = stack\_computer.pop()

cur\_time = stack\_time.pop()

visited.add(cur\_computer.id)

for edge in cur\_computer.edges:

if edge.node.id not in visited:

# calculate time

time = file\_size / edge.rate

stack\_computer.push(edge.node)

if cur\_receive\_rate == float(‘inf’):

time = 0

else:

time = file\_size / (max(cur\_receive\_rate,edge.rate))

stack\_time.push(time + cur\_time)

stack\_receive.push(edge.rate)

max\_time = max(max\_time,time+cur\_time)

return max\_time

Hi:)