# QUESTION 2

j1(size, service rate, tau)

j2(size, service rate, tau) --> tau =time it took from j1 to j2?

(size, service rate, tau)

(size, service rate, tau)

jN(size, service rate, tau)

T\_q, W\_q, COMPUTE BY FORMULA

compare them with the simulated T\_q and W\_q

size --> Pareto Ditribution

tau--> Pareto Ditribution

service Speed --> Exponential

code GG1, (must online stuff),

MM1:

T\_tot = T\_node + T\_queue

GG1:

T\_tot = T\_node + W\_q

e.g.

4MB, 2.3MB

4MB in total,

2.3MB we can process in 1 second

--> it takes 2 cycles of 1 second to compelte a 4MB

# QUESTION 3

1. Draw diagram
2. Write down Q matrix (according to diagram)
3. Compute PI, solve PI \* Q = 0, AND SUM PI = 1
4. E[n] according to PI and POSSIBLE STATES TABLE
5. E[T], usually it’s little’ law, but depents on the schemtics of the diagram
6. Choose what to buy: read the slides +logical reasonig
7. For question3.3 read notes, and check how to compute E[T] for transitions (check Q)

# QUESTION 4

1. Check lecture 8, page 260,
2. Code python just to evaluate the Pq formula

# QUESTION 5

1. Write down 2 cmtc, one for each scenario, then computes Pis and then little’s law

# QUESTION 2

1. Code simulation <https://towardsdatascience.com/simulating-a-queuing-system-in-python-8a7d1151d485>
2. Remove outliers (NOTE THRESHOLD CHECK)  
   # z\_scores = stats.zscore(df\_small)   
   # abs\_z\_scores = np.abs(z\_scores)  
   # filtered\_entries = (abs\_z\_scores < 3).all(axis=1)  
   # new\_df = df\_small[filtered\_entries]  
   # # print(new\_df.head())
3. Check book for formulas