Problem 1 (2.5 points)**. Non-random graph, Cayley Tree (Barabasi 3.11.4)** A Cayley tree is a symmetric tree, constructed starting from a central node of degree k. Each node at distance d from the central node has degree k, until we reach the nodes at distance P that have degree one and are called leaves (see figure for a Cayley tree with $k = 3$ and $P = 5$).

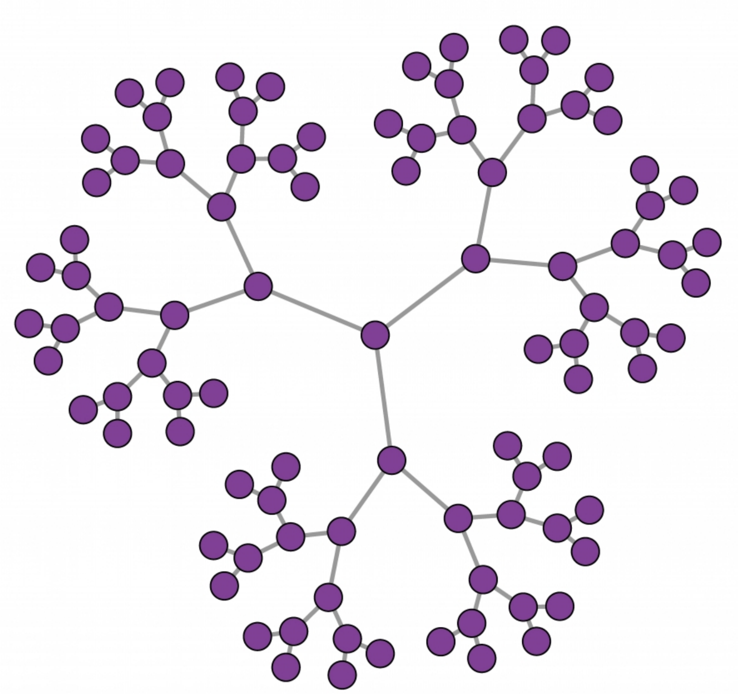

Figure 1: A Caley tree with $k = 3$ and $P = 5$.

1. How many nodes are reachable in t steps from the central node?
2. What is the degree distribution of the network?
3. Calculate the diameter d_{max} of a (k, P) Cayley tree.
4. Express the diameter d_{max} in terms of the total number of nodes N.
5. Does the (k, P) Cayley tree display the small-world property, why or why not?

Problem 2 (5 points)**. The air-traffic network.** We will be using the global air traffic network data, available here: http://seeslab.info/downloads/air-transportation-networks/. Please read the description of the data on the page where you download the data. Node information is stored in `global-cities.dat` (node ids are the second column), and an edge list (on node ids) is stored in `global-net.dat`.

Completing the assignment You may choose to use any programming language. We highly recommend a python notebook with the `networkx` package. You are welcome to use third party packages such as graph libraries, for example networkx in python, igraph in R, and so on – this is strongly recommended.

1. Basic graph attributes: how many nodes and (undirected) edges are in this network? (0.5 point)

2. How many connected components are in this graph, how many nodes and edges do the largest component contain? (0.5 point)

3. Denote the largest component as G. List the top 10 nodes in G having the highest degree, and how many other nodes are they connected to. Please give names of the city/airport, node ids will not be accepted. (0.5 point)

4. Plot the degree distribution of the network G. Each data point is a pair (x,y) where x is a positive integer and y is the fraction of nodes in the network with degree equal to x. Also plot the degree distribution on a log-log scale. Restrict the range of x between the minimum and maximum degrees. You may filter out data points with a 0 entry. For the log-log scale, use base 10 for both x and y axes. Include numeric labels on each axes for the reader to make sense of the plot. (1 point)

5. What is the (unweighted) diameter of the giant component G in this network? List a longest (unweighted) shortest path between two cities. Please give names of the city/airport, node ids will not be accepted. (1 point)

6. What is the smallest number of flights you need to take to get from Canberra (CBR) to Cape Town (CPT)? Which airports does your route take you through? Please give names of the city/airport, node ids will not be accepted. (1 point)

7. Which airport/city in G is most "central" by having the largest betweeness, list the top 10 cities with their betweeness value? (0.5 point)

Problem 3 (2.5 points). **Small component in $G(N, p)$.** Consider Erdös-Rényi random graph $G(N, p)$ in a supercritical state, i.e. $Np > 1$ with large enough N. Given a small component consisting of s nodes that is not part of the giant component, show that it is likely to be a tree. Noting that a tree consisting of s nodes has exactly $s - 1$ edges.

1. Compute the expected number of edges in addition to the $s - 1$ edges.

2. Denote the expected node degree $< k >= (N - 1)p$. Argue that there is at least some nodes with less than $< k >$ connections in this small component. Consider adding one new edge to such a node, what is the probability that such a new edge is within the small component, or to the giant component?

Problem 4 (5 points). Consider a set of nodes, which represent individuals in a social network, that are placed on lattice points in an $n \times n$ square $\mathcal{N} = \{(i, j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\}\}$. We define the lattice distance between two nodes (i, j) and (k, l) to be the number of "lattice steps" separating them: $d((i, j), (k, l)) = |k - i| + |l - j|$. Suppose every node has a directed edge to every other node within lattice distance 1 (local-contacts). For each node u, we also construct **one** directed edge to another node as its long-range contact: The probability that there is a directed edge from u to another node v is proportional to $[d(u,v)]^{-2}$. For two arbitrary nodes s and t, we want to deliver a message from s to t through the contacts. We assume the message holder in a given step knows of

- The set of local contacts among all nodes (i.e. the underlying grid structure);
- The location, on the lattice, of the target t.
- Its own long-range contact.

Prove (by contruction) that there is a decentralised algorithm to deliver the message within $\mathcal{O}((\log(n))^2)$ steps. Similar to the 1-dimensional case in EK book Chapter 20, one can use the following three-step breakdown to complete the proof.

1. Fix an arbitrary node u, show that $\sum_{v \in \mathcal{N}, v \neq u} d(u,v)^{-2} \leq 4\ln(6n)$. (Hint: Convert the LHS into a summation over the lattice distances of contacts - what should be the longest distance? For a given distance j, show that there are at most $4j$ nodes that are j steps away from u.)

2. The decentralized algorithm is defined as follows: in each step, the current message holder u chooses a contact that is as close to the target t as possible, in the sense of lattice distance. For $j > 0$, we say that the execution of the algorithm is in phase j when the lattice distance from the current node to t is greater than 2^j and at most 2^{j+1}. Let the random variable X_j denote the total number of steps spent in phase j (i.e. The total number of steps to enter phase $j-1$). Show that $\mathbb{E}[X_j] \leq 128\ln(6n)$. (Hint: Fix some j, derive a lower bound of the number of nodes with lattice distance at most 2^j to t and an upper bound of the lattice distance between the current node and t. Then, what is the lower bound of the probability that we can enter into the next phase through a long contact? To get the upper bound of the expectation, thinking about geometric distribution might be helpful.)

3. Conclude the proof: Show that the algorithm could deliver the message within $\mathcal{O}((\log_2(n))^2)$ steps in expectation.