Foundations of Algorithms, Spring 23
Homework #2

All members of the collaboration group are expected to participate fully in solving collaborative problems. Note, however, that each student is required to write up their solutions individually. Common solution descriptions from a collaboration group will not be accepted. Furthermore, to receive credit for a collaboration problem, each student in the collaboration group must actively and substantially contribute to the collaboration. This implies that no single student should post a complete solution to any problem at the beginning of the collaboration process.

Problems for Grading

1. [30 points] Give an $O(n \lg k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (*Hints*: Use a heap for k-way merging.)

2. [30 points] Suppose that instead of swapping element $A[i]$ with a random element from the subarray $A[i...n]$, we swapped it with a random element from anywhere in the array.

Algorithm 1. Permute with All

```
function PermuteWithAll(A)
    n ← A.length
    for i ← 1 to n do
        swap A[i] with A[Random(1,n)]
```

Does this code produce a uniform random permutation? Why or why not?

3. [40 points] ***Collaborative Problem***: A number of *peer-to-peer systems* on the internet are based on *overlay networks*. Rather than using the physical internet as the network on which to perform computation, these systems run protocols by which nodes choose collections of virtual "neighbors" so as to define a higher-level graph whose structure may bear little or no relation to the underlying physical network. Such an overlay network is then used for sharing data and services, and it can be extremely flexible compared with a physical network, which is hard to modify in real time to adapt to changing conditions.

   Peer-to-peer networks tend to grow through the arrival of new participants who join by linking into the existing structure. This growth process has an intrinsic effect on the characteristics of the overall network. Recently, people have investigated simple abstract models for network growth that might provide insight into the way such processes behave in real networks at a qualitative level.

   Here is a simple example of such a model. The system begins with a single node v_1. Nodes then join one at a time; as each node joins, it executed a protocol whereby it forms a directed link to a single other node chosen uniformly at random form those already in the system. More concretely, if the system already contains nodes $v_1, ..., v_{k-1}$ and node v_k wishes to join, it randomly selects one of $v_1, ..., v_{k-1}$ and links to that node.

   Suppose we run this process until we have a system consisting of nodes $v_1, ..., v_n$; the random process described above will produce a directed network in which each node other than v_1 has exactly one outgoing edge. On the other hand, a node may have multiple incoming links, or none at all. The incoming links to a node v_j reflect all the other nodes whose access into the system is via v_j; so if v_j has many incoming links, this can place a large load on it. Then to keep the system load-balanced, we would like all the nodes to have a roughly comparable number of incoming links. That is unlikely to happen, however, since nodes that join earlier in the process are likely to have more incoming links than nodes that join later. Let us try to quantify this imbalance as follows.

   (a) [20 points] Given the random process described above, what is the expected number of incoming links to node v_j in the resulting network? Give an exact formula in terms of n and j, and also try to express this quantity asymptotically (via an expression without large summations) using $\Theta(\cdot)$ notation.

   (b) [20 points] Part (a) makes precise a sense in which the nodes that arrive early carry an "unfair" share of connections in the network. Another way to quantify the imbalance is to observe that, in a run of this random process, we expect many nodes to end up with no incoming links. Give a formula for the expected number of nodes with no incoming links in a network grown randomly according to this model.