Problem 19 (4 points):
Show that there is a caching strategy for the hypercube that is $O(\log n)$ competitive. (Hints: Use a proof similar to the one for the mesh. Recall the oblivious routing strategy we used for the hypercube and the fact that a $d$-dimensional hypercube contains two $d-1$-dimensional hypercubes.)

Problem 20 (4 points):
Consider the following strategy: Given a constant-degree network $G = (V, E)$ with $n$ nodes and flow number $F$, simply take a complete binary tree $T(G)$ with the nodes of $G$ as leaves for each object and map each node of $T(G)$ randomly to a node in $G$. Show that in this case the competitive ratio of the caching strategy is $O(F \log n)$. 