1 Concurrent Ivy

Consider the tree for the Ivy shared variable protocol in Figure 1. There are three concurrent requests placed by the nodes \( v_1, v_2 \) and \( v_3 \). The token is initially held by the circled node labeled \( r \). We assume synchronous execution.

a) Give the order of serviced requests.

b) Draw the tree after the last request has been served.

c*) Show that in an asynchronous setting, Ivy incurs at most an \( O(\log n) \) overhead in amortized message complexity.

2 Tight Ivy

In Theorem 6.5 it was shown that, on average, acquiring a lock requires at most \( \log n \) steps, where \( n \) is the number of processors.

Show that this bound on the number of steps is tight by constructing a tree consisting of \( n \) nodes in which each request requires \( \log n \) steps if all requests are performed sequentially by suitable nodes in the tree.\(^1\)

\(^1\)Hints: Assume that \( n \) is a power of 2. Construct a tree whose topology remains the same with respect to the token holder after each request.