def train(\mathcal{D}): 
    store root = tree_recurse(\mathcal{D})

def tree_recurse(\mathcal{D}'): 
    q = new node() 
    base case – if (SOME CONDITION): 
    recursion – else: 
        q.type = internal 
        find best attribute to split on, x_d 
        q.split = x_d 
        for v in V(x_d), all possible values of x_d: 
            \mathcal{D}_v = \{(x^{(n)},y^{(n)}) \in \mathcal{D} | x_d^{(n)} = v\} 
            q.children(v) = tree_recurse(\mathcal{D}_v) 
    return q
def train($\mathcal{D}$):
    store root = tree_recurse($\mathcal{D}$)

def tree_recurse($\mathcal{D}'$):
    q = new node()
    base case – if ($\mathcal{D}'$ is empty OR all labels in $\mathcal{D}'$ are the same OR all features in $\mathcal{D}'$ are identical OR some other stopping criterion):
        q.type = leaf
        q.label = majority_vote(labels in $\mathcal{D}'$)
    recursion – else:
    return q