Exercise 7.2

maandag 2 oktober 2023 17:15

(i) The height of a binary tree with n nodes is $\log_2 n$. Consider a root-to-leaf path, which must have length $(\log_2 n) - 1$. Now, assume this path visits fewer than $\log_B n = \frac{\log_2 n}{\log_2 B}$ blocks. This would imply that, at some point on the path, there exists a sub-path (say $v_1, v_2, \ldots, v_{m-1}, v_m$) where more than $\log_2 B$ nodes are in the same block. Then, we have that the subtree of size B rooted in v_1 (which has height $\log_2 B$) must contain at least one node which is not part of the same block. But that implies that any root-to-leaf path going through that node would use fewer than $\log_2 B$ nodes from some block (given that blocks cannot be re-entered). Given that *the same reasoning* can also be applied to any paths going through this node, we can then conclude that there must exist at least one path where fewer than $\log_2 B$ vertices from a block are used on average. This suggests that the number of blocks on that path must be greater than $\log_B n = \frac{\log_2 n}{\log_2 B}$ (which proves the claim).

a. Honestly, I am not convinced at this point.

(ii) ~~Assume that the condition in (i) is not met; that is, we have that there exists a root-to-leaf path which leaves a block and later re-enters the same block. Then, we modify the blocking strategy as follows:~~

a. ~~Consider the node where the block is re-entered; we swap this node from the block for the node where the block was first left.~~

b. ~~Repeat this procedure until no blocks are re-entered anymore.~~

~~By applying this strategy, the number of visited blocks does not increase; given that any path which goes through the node where the formerly re-entered block first left its block already passed through the formerly re-entered block, we have that the number of blocks on any path going through the formerly re-entered block cannot grow.~~

(iii) Assume that the condition in (i) is not met; that is, we have that the nodes in at least one block do not form a connected component. Then, we can modify the blocking as follows:

Attempt 2

(i) We only use the implication of the given property: whenever a root-to-leaf path leaves a block, it will not re-enter that block. *(Although this is not necessary for this part of the exercise, limiting ourselves to this assumption simplifies the proof for the second question.)* We prove by induction that, if the blocking strategy is such that this condition is met, then there is always a root-to-leaf path which visits at least $\frac{1}{2}\log_{B+1} n$ blocks. Note that this suffices to prove that any path visits $\Omega(\log_B n)$ blocks, since $\frac{1}{2}\log_{B+1} n = \frac{1}{2}\frac{\log_B n}{\log_B B+1} = \Omega(\log_B n)$.

Base case ($n \leq B$)

In this case, we have that we must visit at least one block, whereas $\frac{1}{2}\log_{B+1} n = \frac{1}{2}\frac{\log_B n}{\log_B B+1} \leq 1$ (given that $\log_B n \leq 1$ for $n \leq B$, and $\frac{1}{2\log_B B+1} < 1$). Hence, the claim holds.

Inductive step ($n > B$)

In this case, we start by considering that every path from the root to a leaf must at least visit the block which contains the root. Furthermore, we have that there are $n - B$ nodes outside this block, and these are split over (at most) $B + 1$ subtrees (since there are B nodes in the block containing the root, and the number of edges connecting these nodes to a node outside the block is at most $B + 1$). This means that, on average, these subtrees have $\frac{n-B}{B+1}$ nodes. If the subtrees contain at least this many nodes on average, then there must also be at least one subtree which contains this many nodes. We now consider the paths going to a leaf in this subtree. By the induction hypothesis, at least one of these paths must visit at least $\frac{1}{2}\log_{B+1}\left(\frac{n-B}{B+1}\right)$ blocks, plus one additional block to account for the block containing the root node of the tree. Thus, at least one of these paths visits at least $1 + \frac{1}{2}\log_{B+1}\left(\frac{n-B}{B+1}\right) = 1 + \frac{1}{2}(\log_{B+1}(n-B) - \log_{B+1}(B+1)) = 1 + \frac{1}{2}(\log_{B+1}(n-B) - 1) = \frac{1}{2} + \frac{1}{2}\log_{B+1}(n-B) = \frac{1}{2}(\log_{B+1}(n-B) + 1) = \frac{1}{2}(\log_{B+1}(n-B) + \log_{B+1}(B+1)) = \frac{1}{2}\log_{B+1}((n-B)(B+1)) = \frac{1}{2}\log_{B+1}(nB - B^2 + n - B) \geq \frac{1}{2}\log_{B+1} n$, where the last step holds because of the following:

- n and B are both integers, and $n > B$, so we have $n \geq B + 1$
- Then, $nB \geq (B+1)B = B^2 + B$
- Hence, we have $nB - B^2 - B \geq B^2 + B - B^2 - B = 0$
- Thus, $nB - B^2 + n - B \geq n$.

This proves the claim.

Hence, it must be the case that for a blocking strategy which meets the condition laid out at the top, there is always a root-to-leaf path which visits at least $\Omega(\log_B n)$ blocks.

(ii) To see why there must be a root-to-leaf path that visits $\Omega(\log_B n)$ blocks for any blocking strategy, we consider that any blocking strategy which does not satisfy the property used in the proof of the first question must have the following:

There must be nodes u, v, and w, such that v is a child u, w is a descendant of v, u and w are in the same block, and v is not in the same block as u and w.

We note that, in this situation, it is possible to swap the nodes v and w; that is, we can add node w to the block v was in, while adding node v to the block w was in. This does not increase the number of blocks visited by any root-to-leaf path going through u, as, in the new situation, we have that paths going through u, v and w (or which do not visit either of v and w) visit an equal number of blocks, whereas paths not going through v might visit fewer (but not more) blocks. Furthermore, performing this swap leads to a blocking strategy which is closer to one satisfying the property from the first question. Hence, it is always possible to transform a grouping of nodes into blocks to one satisfying the property from the first question without increasing the number of blocks on any root-to-leaf path, and hence, the number of visited blocks on any root-to-leaf path in a blocking strategy which does not satisfy the property must be at least as large as the number of visited blocks in a blocking strategy which does satisfy the property (i.e. $\Omega(\log_B n)$). This implies that any blocking strategy must have a root-to-leaf path which visists at least $\Omega(\log_B n)$ blocks.