cmp - solution

There are several solutions for this problem. The best solution the Scientific Committee has uses T = 10 , but we believe that T = 9 is also achievable and deserves more than 100 points.

- *A simple T = 17 solution:*

We consider a binary tree that has 4096 leaves.

Remember(a): we set all the nodes from leaf “a” to the root to 1.

maxA = 12

Compare(b): Our goal is to determine the lowest common ancestor (LCA) for leaf a and leaf b. Since the whole path from the root to the leaf is set to 1 we can do a binary search. Once we have the LCA we can return the -1, 0 or 1.

maxB = 5

The next solutions use **one-hot encoding**.

This encoding uses N bits for representing a number from 0 to N-1. For a given number X we only set the bit X to 1 and leave the rest set to 0. The advantage of this method is that we only need to write 1 bit. While reading a value could take N – 1 reads (we look for the bit set to 1), comparing is a little faster. It takes (N-1)/2 reads. We only need to search from the current position to the closest end (0 or N-1).

- *Some tree solutions (T = 13):*

We change the simple binary tree from the previous solution. We still have at least 4096 leaves, but the internal nodes will have a variable number of children. For instance let's assume that we have 6 levels and the number of children for each level is A, B, C, D, E and F. We chose the degree for each level such that A*B*C*D*E*F >= 4096.

To encode a value in a node we use the **one-hot** encoding.

Let's consider A = B = C = D = E = F = 4

Remember(a): we set all the nodes from leaf “a” to the root to 1.

maxA = 6, since we have 6 nodes and encoding takes 1 bit_set() call.

Compare(b): Considering the same representation for b again we look for LCA. However this time we do a top-down traversal. As soon as the bit we expect to be set to 1 is 0 we try to determine if b is higher or lower.

Worst case is when the last bit is not set 1. In this case we need to read an additional bit.

maxB = 7, since we have 6 nodes and an additional *bit_get*().

- *Further optimizations (T = 12):*

For the previous solution we observe that the worst case is achieved when we miss the bit in the last node.

If we choose A = 7, B = 6, C = 5, D = 4, E =3, F = 2 then maxB becomes 6. This solution also works if comparing a one-hot encoding is done in N – 1 reads and not (N-1)/2.

- *Breaking the maxA – maxB symmetry (T = 10):*

Instead of 6 levels for our tree we chose 4 levels of degree: 12, 10, 8, 6.

maxA becomes 4 and maxB becomes 6.

This solution uses the T = 12 optimization.

For a given set of maxim branching factors the formula for T is:

num_levels + max(branching_factor_on_level[i]/2 + 1 + i for i in range(0, num_levels))

with *product(branching_factor_on_level(i)) > 4095*

- *Mixed-radix*

All the above tree solutions don't really need a tree. For instance let's pick the T = 10 above solution.

We can encode the value from the root in the bits from 1 to 12, the value from the node on the second level in the memory from 13 to 23 and so on.