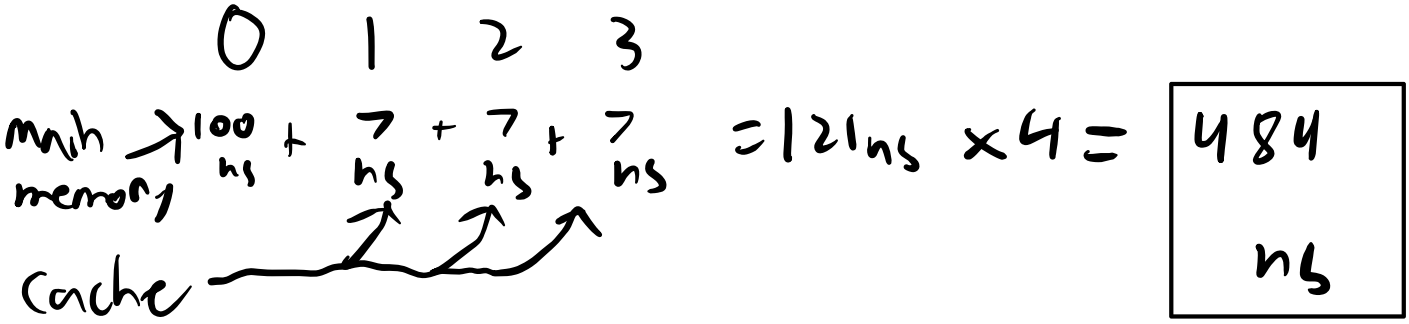
1.) linked list: All main memory

$16 \times 100 =$ **1,600 ns**

vector:



$= 121 \text{ ns} \times 4 =$ **484 ns**

2.)

(a.) This could definatly be parralized

Steps: divide linked list into sections based on the number of processors / threads

2. Push the largest from each section into a Queue

3.) return max(Queue)

b.) I dont think this should be parralized unless the stack can support pushing multiple things at once and the order of the stack does not matter. If only one element can be pushed at a time, than a mutex would be needed, and at that point it is basically serial.

C.) I dont think this should be parralized because of the unique structures of binary search trees. When searching these trees there is only one possible path from the head to the wanted node, so multiple threads are kinda pointless.