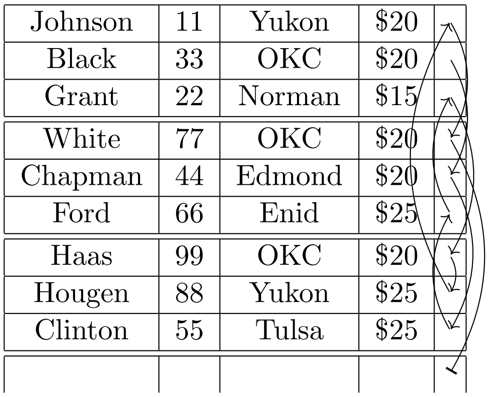
NAME: Nicholas Jacob
STUDENT ID: # 113578513
GRADED HOMEWORK NUMBER: 3 Question 3
COURSE: CS/DSA 4513 DATABASE MANAGEMENT
SECTION: ONLINE
SEMESTER: FALL 2023
INSTRUCTOR: DR. LE GRUENWALD
SCORE:

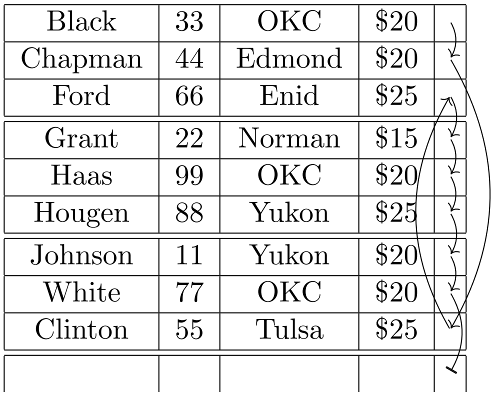
1. First I will consider that the records have freshly been inserted and no reordering on the physical system has been re-done. This will require pointers going every which way... The first pointer goes to Black...



Johnson	11	Yukon	$20	
Black	33	OKC	$20	
Grant	22	Norman	$15	
White	77	OKC	$20	
Chapman	44	Edmond	$20	
Ford	66	Enid	$25	
Haas	99	OKC	$20	
Hougen	88	Yukon	$25	
Clinton	55	Tulsa	$25	

I could not find the exact arrow used in notes for the last pointer. I am terminating by having a null pointer.

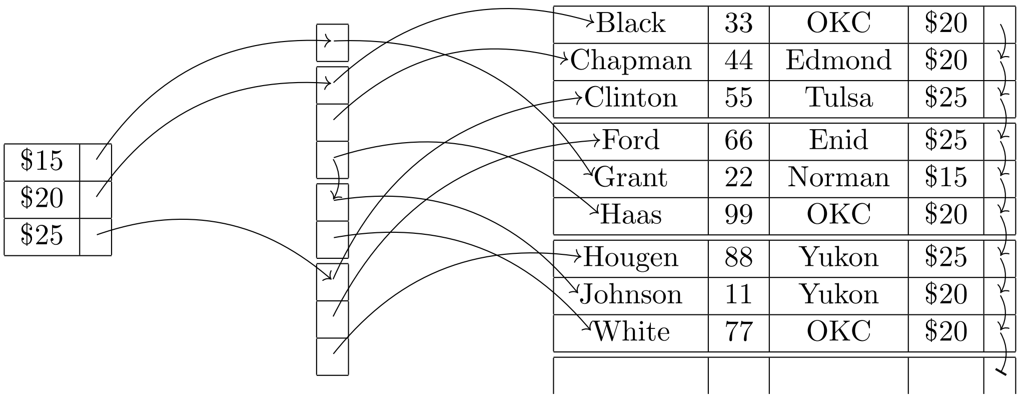
So clearly this is the result of all those inserts in a row. We know that the system will rearrange sequential files periodically so let's do that and have the last file be an insert.



Black	33	OKC	$20	
Chapman	44	Edmond	$20	
Ford	66	Enid	$25	
Grant	22	Norman	$15	
Haas	99	OKC	$20	
Hougen	88	Yukon	$25	
Johnson	11	Yukon	$20	
White	77	OKC	$20	
Clinton	55	Tulsa	$25	

That is somewhat less of a mess...

2. For the second question, I assume the index sequentially by name is full cleaned.



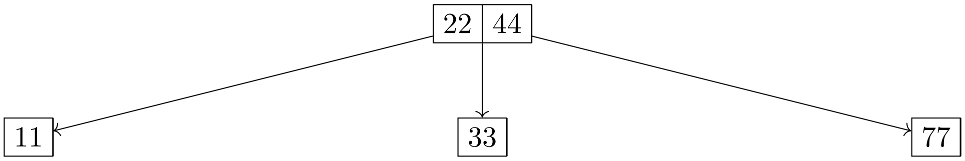
$15	
$20	
$25	

Black	33	OKC	$20	
Chapman	44	Edmond	$20	
Clinton	55	Tulsa	$25	
Ford	66	Enid	$25	
Grant	22	Norman	$15	
Haas	99	OKC	$20	
Hougen	88	Yukon	$25	
Johnson	11	Yukon	$20	
White	77	OKC	$20	

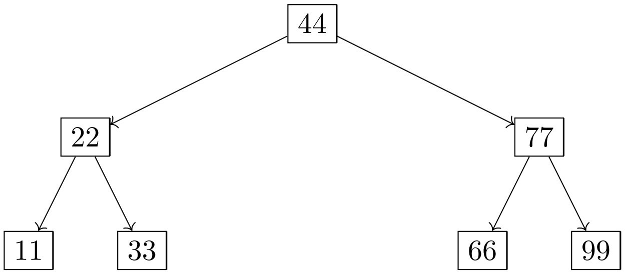
So not a terrible mess (although you should see the LaTeX code it took to generate this...) Highlights include pointers from each dollar amount into sequentially stored pointer of the data (I even kept the three requirement on the $20 breaking that into two blocks)

3. For creating the B tree of order 3, we will have three pointers to next part of tree available at each block and room for two indexes. I will follow the algorithm outlined in the notes.

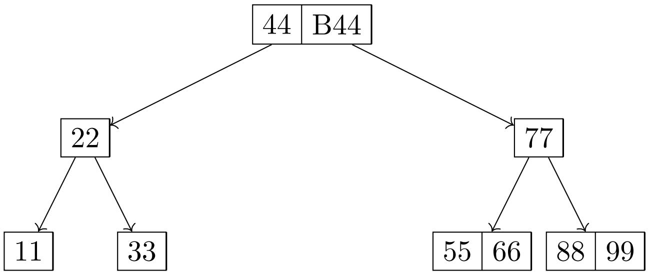
   - First two insertions are trivial. Just filling the root. $[11, 33]$.
   - As 22 is inserted, we have filled the first block so we get 22 as the root and both 11 and 33 end up in their own blocks.
   - 77 joins 33. No issues
   - 44 Will fill that block though so 44 will need to go to the parent. Let's make that picture!



   - 66 will have a spot but 99 fills that node so 77 will go to the root which fills the root too. This must split again. 44 is the middle value it becomes the root. Another picture is in order.



- 88 and 55 have slots in the leaves.


44
B44
22
77
11
33
55
66
88
99