Chris Banci

CS443

May 7, 2017

**CS443 -- Lab 5**

**Question 1:**

A PARTS file with Part# as hash key includes records with the following Part# values: 2369, 3760, 4692, 4871, 5659, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 6975, 4981, 9208. The file uses 8 buckets, numbered 0 to 7. Each bucket is one disk block and holds two records. Load these records into the file in the given order using the hash function h(K)=K mod 8. Calculate the average number of block accesses for a random retrieval on Part#.

**Hashing:**

h(2369) = 2369 % 8 = 1

h(3760) = 3760 % 8 = 0

h(4692) = 4692 % 8 = 4

h(4871) = 4871 % 8 = 7

h(5659) = 5659 % 8 = 3

h(1821) = 1821 % 8 = 5

h(1074) = 1074 % 8 = 2

h(7115) = 7115 % 8 = 3

h(1620) = 1620 % 8 = 4

h(2428) = 2428 % 8 = 4

h(3943) = 3943 % 8 = 7

h(4750) = 4750 % 8 = 6

h(6975) = 6975 % 8 = 7

h(4981) = 4981 % 8 = 5

h(9208) = 9208 % 8 = 0

**Buckets:**

bucket 0 = | 3760, 9208 |

bucket 1 = | 2369|

bucket 2 = | 1074|

bucket 3 = | 5659, 7115 |

bucket 4 = | 4692, 1620 | overflow = 2428 |

bucket 5 = | 4981 |

bucket 6 = | 4750 |

bucket 7 = | 4871, 3943 | overflow = 6975 |

**Average:**

13 records can be found with 1 block access

2 records in overflow can be found with 2 block access.

P(records found with one block access) + P(records found with two block access)

* ((13/15) \* 1) + ((2/15) \* 2) = 1.13 block accesses

**Question 2:**

Consider the following records of question 1:

2369, 3760, 4692, 4871, 5659, 1821, 1074, 7151, 1620

Load the records into expandable hash files based on extendible hashing. You can put two records per block. Show the directory at each step, and the global and local depths. Use the hash function h(k) = K mod 32 to find the bucket number. For example,

***Record# K \_ h(K) bucket number binary h(K)***

record1 2369 1 00001

record2 3760 16 10000

record3 4692 20 10100

record4 4871 4871 % 32 = 7 00111

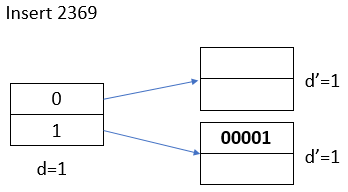
record5 5659 5659 % 32 = 27 11011

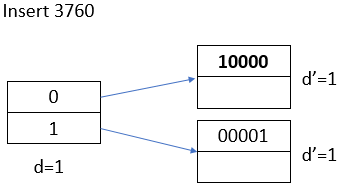
record6 1821 1821 % 32 = 29 11101

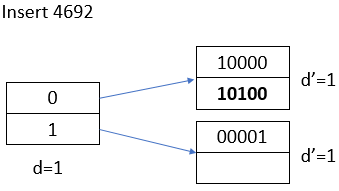
record7 1074 1074 % 32 = 18 10010

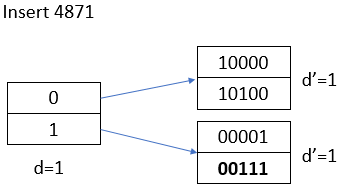
record8 7151 7151 % 32 = 15 01111

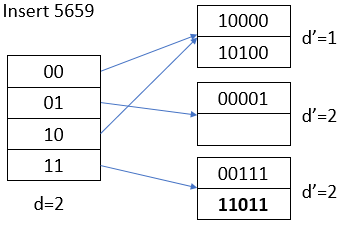
record9 1620 1620 % 32 = 20 10100

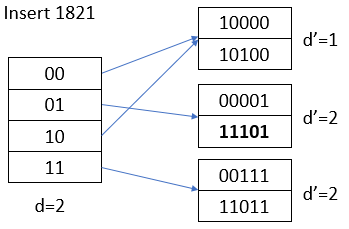


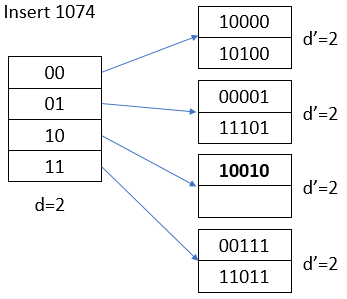


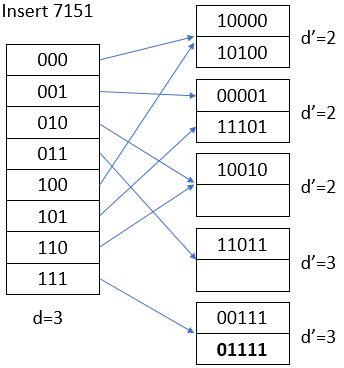


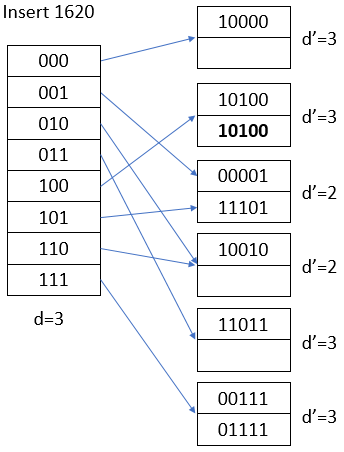












**Question 3:**

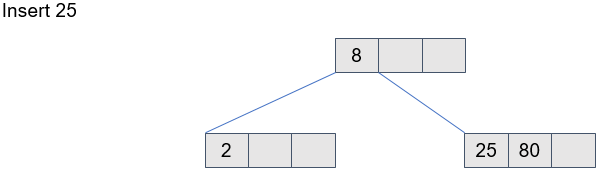
Insert the following into B-tree of order **4**. Show your work step by step with proper illustration of pointers as shown in pages 20-29 in multi-way trees lecture

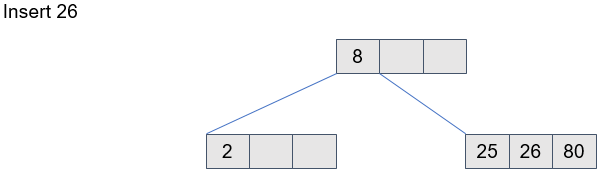
**8, 2, 80, 25, 26, 27, 28, 55, 71, 15, 51, 20, 21, 22, 23, 24, 63, 90, 35**

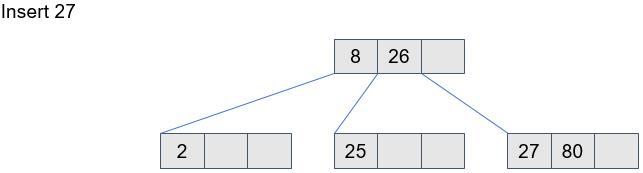


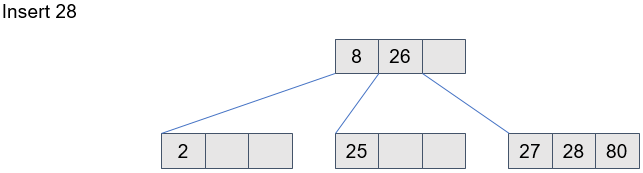


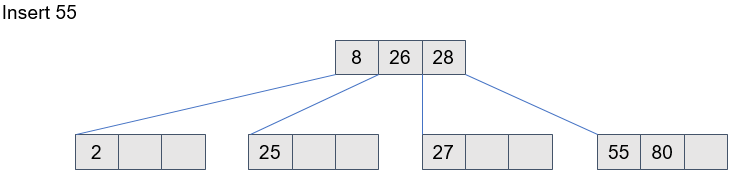


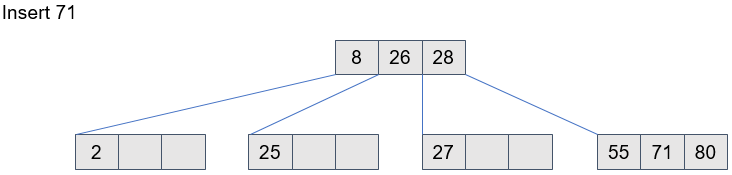


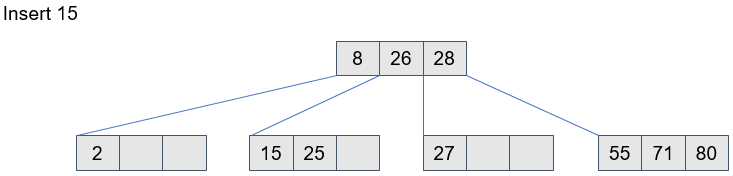


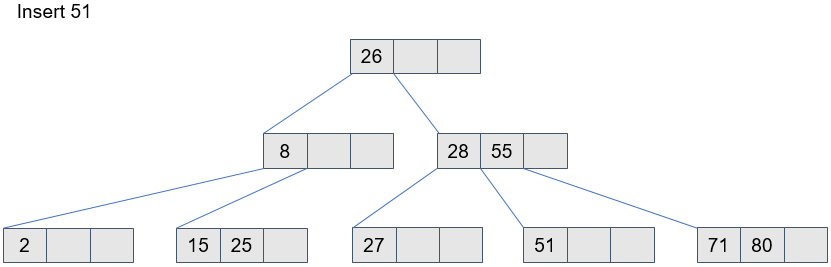


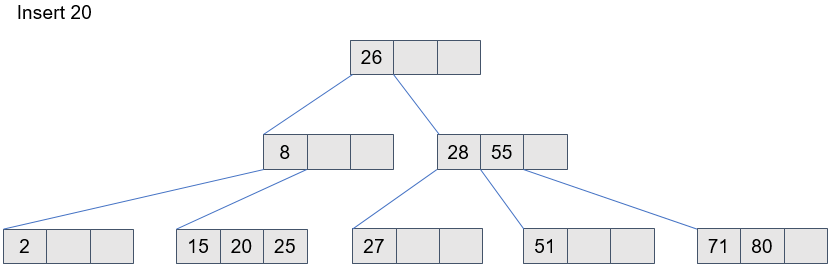


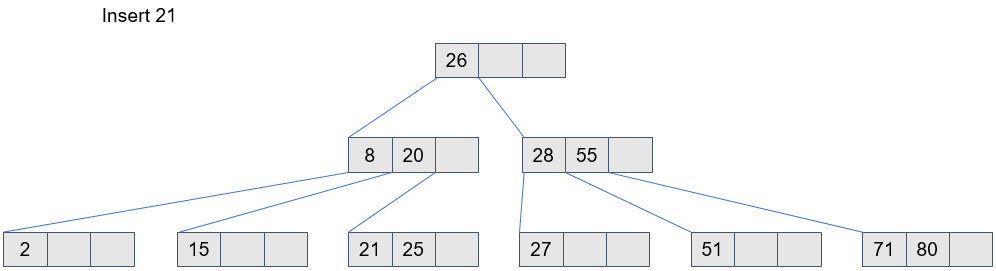


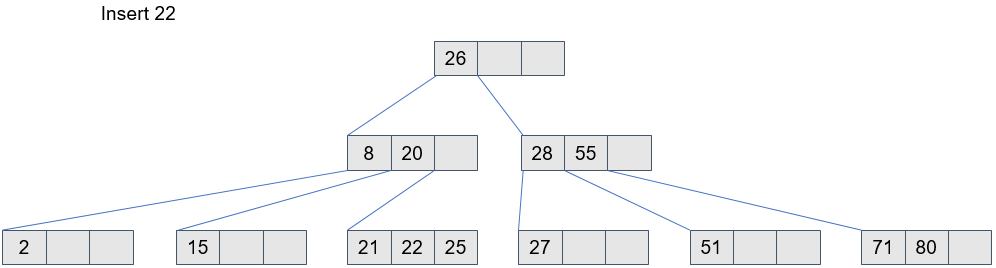


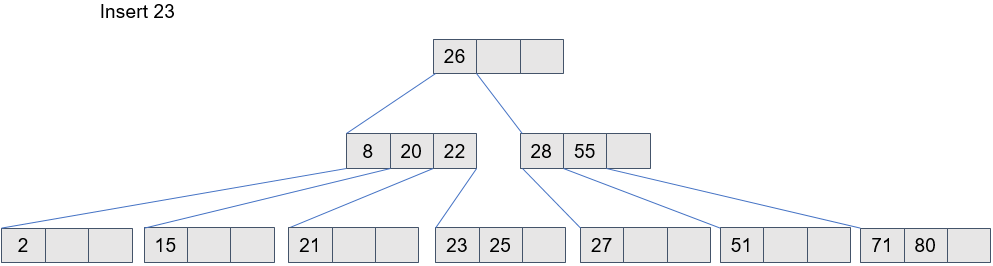


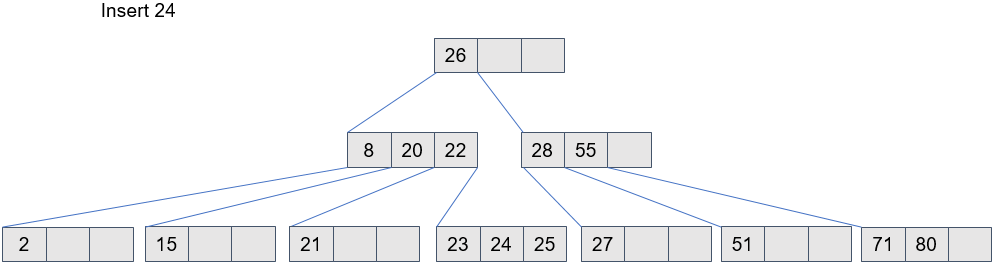


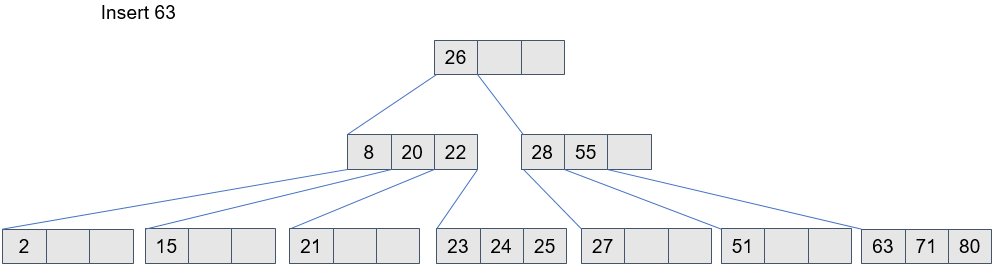


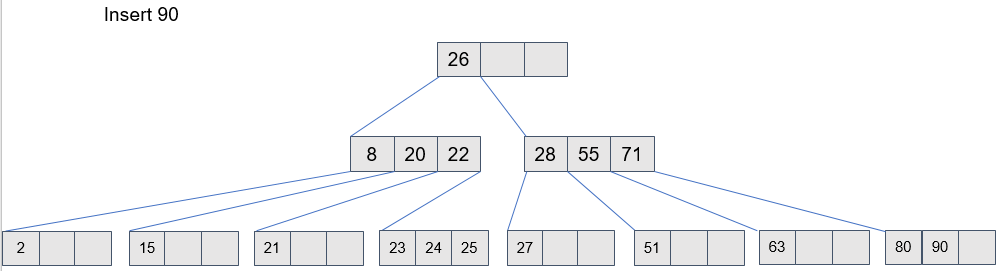


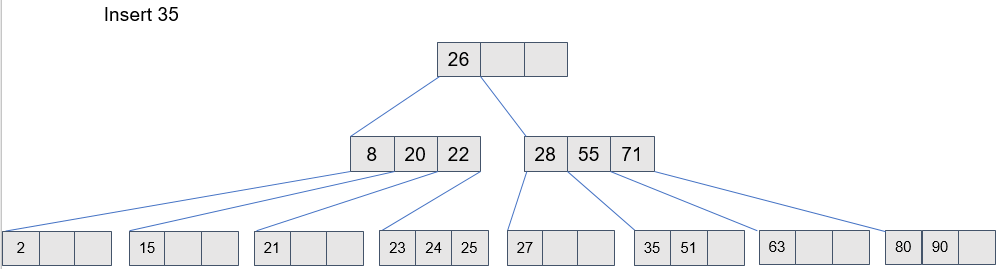












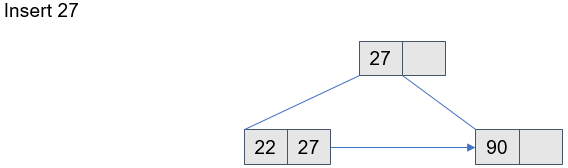
**Question 4:**

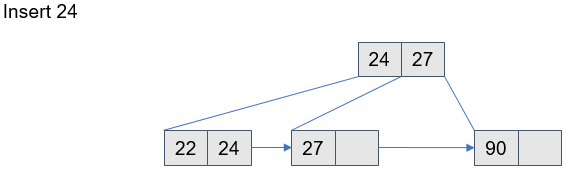
Insert the following into B+ tree of order **3**. Show your work step by step with proper illustration of pointers as shown in pages 47-54 in multi-way trees lecture

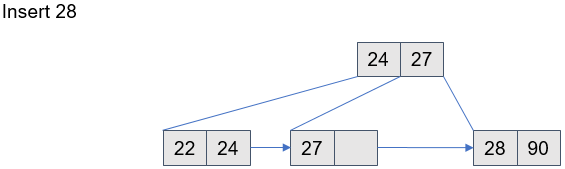
**90, 22, 27, 24, 28, 20, 51, 63, 8, 80, 15, 71, 35, 55**

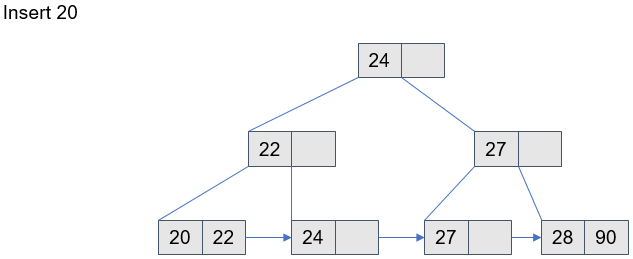


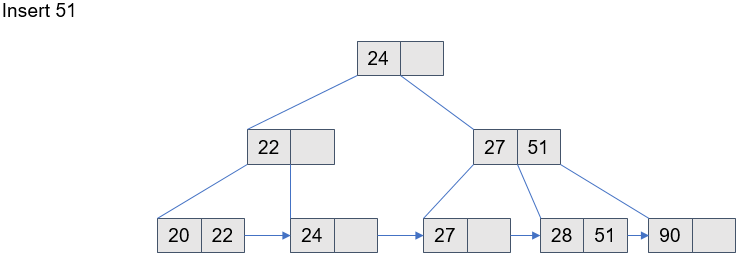


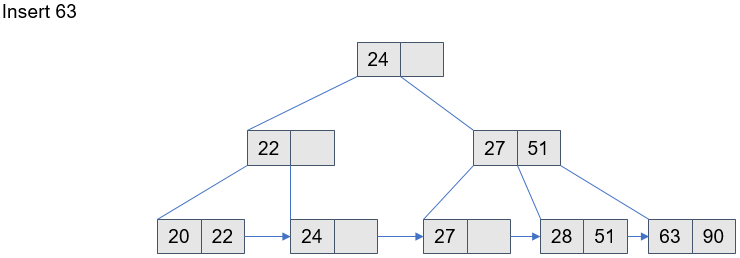


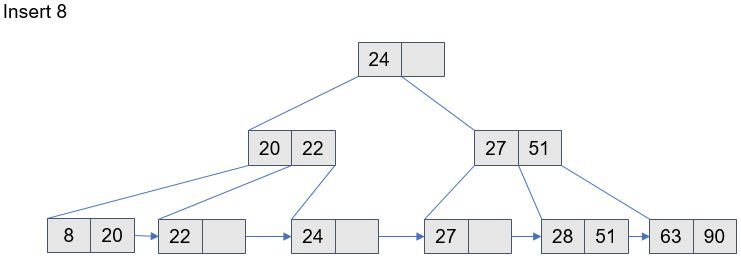


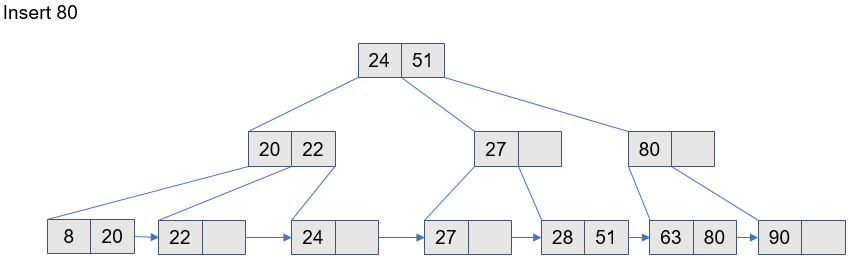


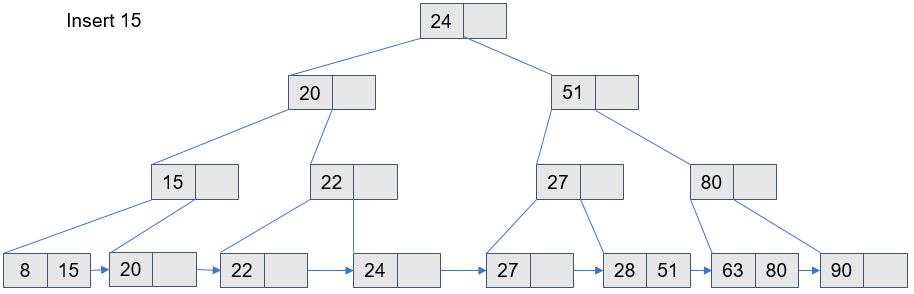


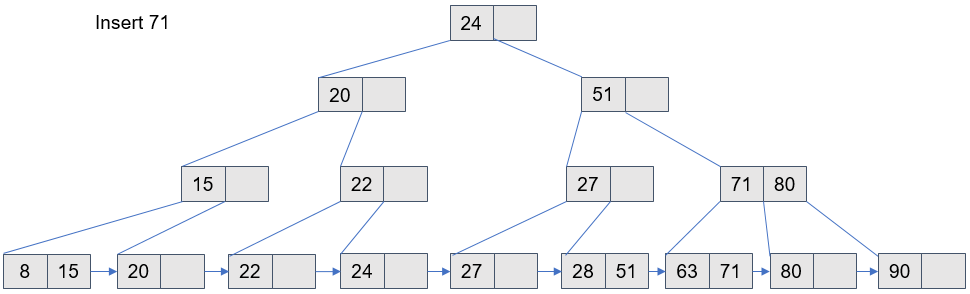


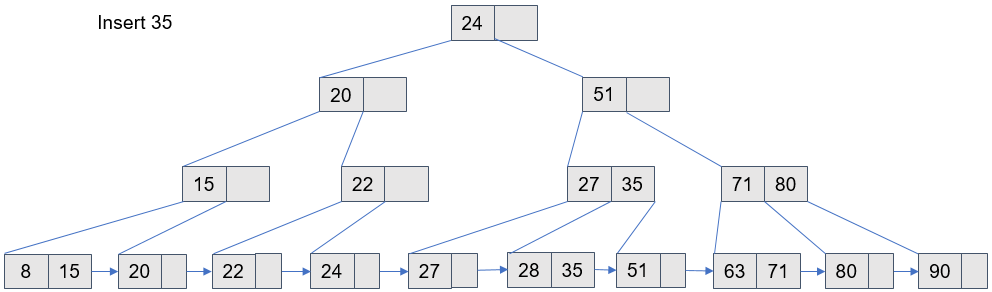


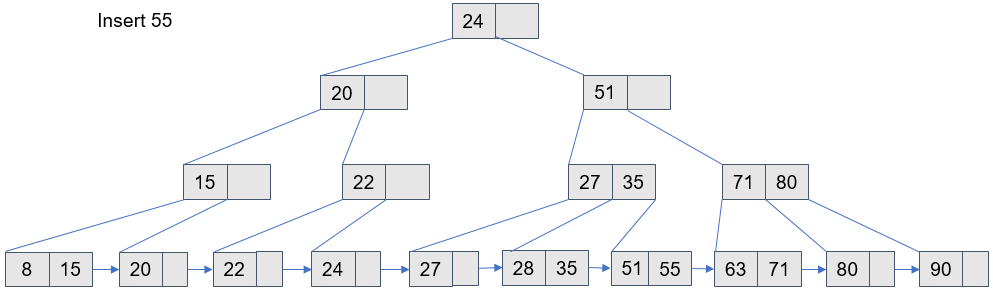






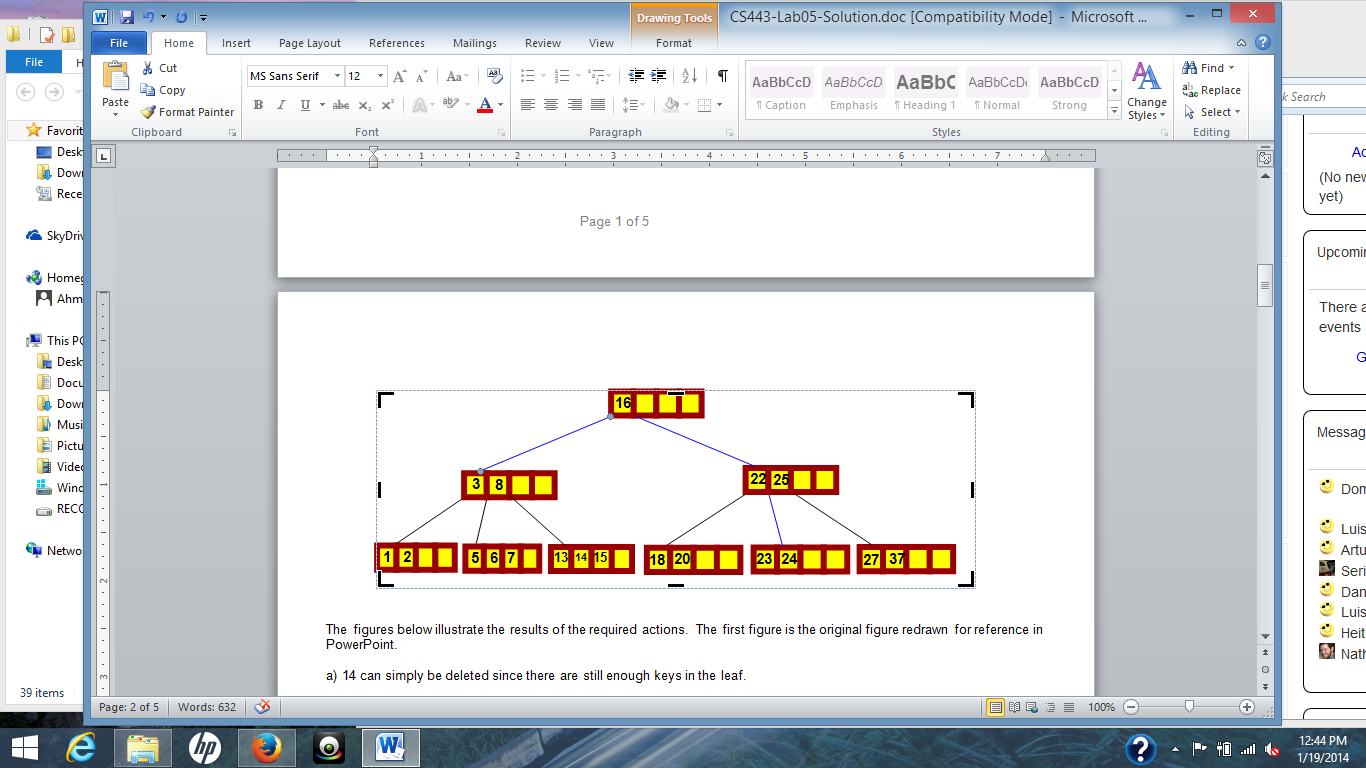


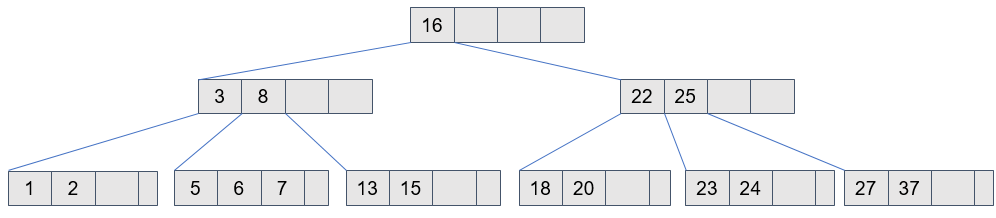




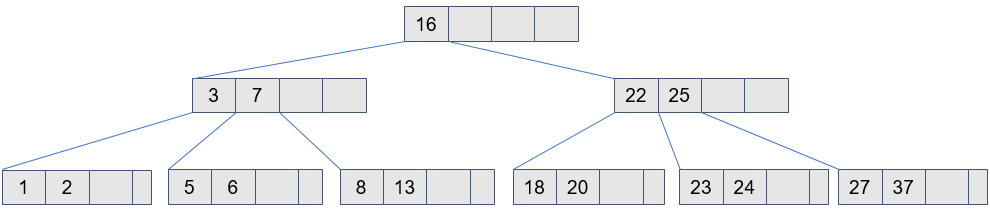
**Question 5:**

Consider the following B-tree.

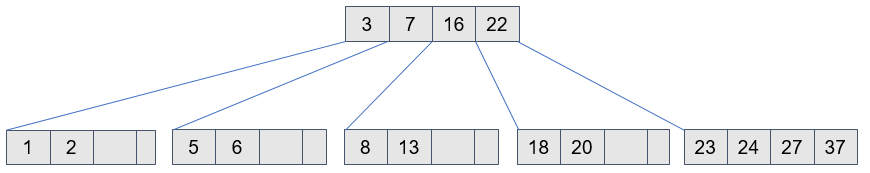


a) Redraw the tree after deleting 14.

b) Again, redraw the tree after deleting 15.

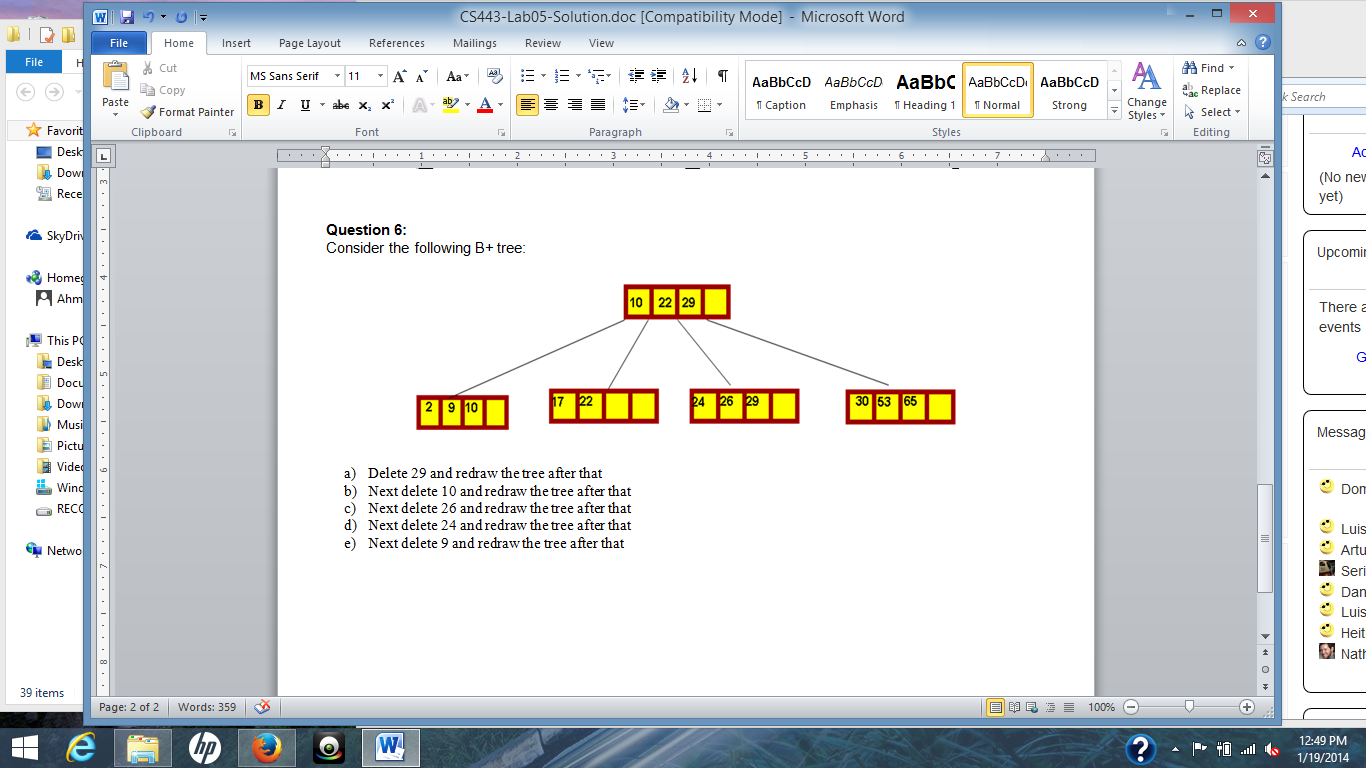


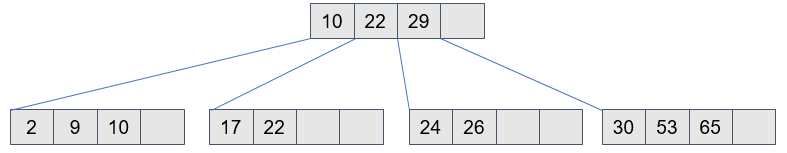
c) Again, redraw the tree after deleting 25.

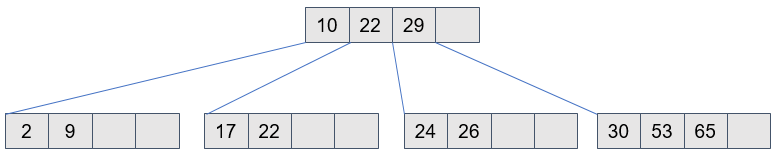


**Question 6:**

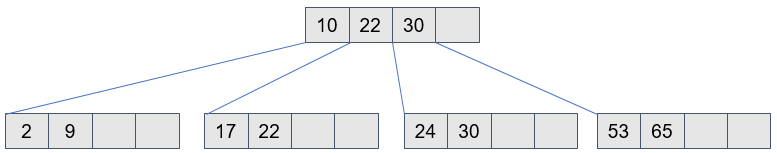
Consider the following B+ tree:



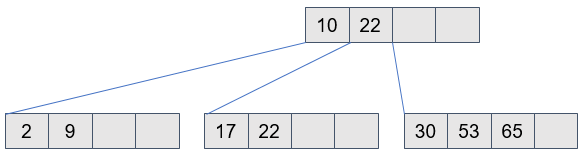
1. Delete 29 and redraw the tree after that
2. Next delete 10 and redraw the tree after that



1. Next delete 26 and redraw the tree after that



1. Next delete 24 and redraw the tree after that



1. Next delete 9 and redraw the tree after that

