FIT3155: Week 8 Tutorial - Answer Sheet

(Scribe: Dinithi Sumanaweera)

Question 2

For any B-tree with the (branching) parameter t and containing n elements overall, work out the minimum possible height of the B-tree

The shallowest B-tree (of which the height is the minimum possible the tree could get) occurs when n elements are arranged in the tree structure such that each node has $(2t-1)$ (= the maximum number of) elements. Let the tree height be h, and consider the number of elements at each height of the tree.

Height/level	# of nodes	# of elements
0	1	$(2t)^0\ (2t-1)$
1	$(2t)$	$(2t)\ (2t-1)$
2	$(2t)^2$	$(2t)^2\ (2t-1)$
3	$(2t)^3$	$(2t)^3\ (2t-1)$
.	.	.
.	.	.
.	.	.
h	$(2t)^h$	$(2t)^h\ (2t-1)$

Accordingly, the total number of elements n can be expressed as:

$$
\begin{aligned}
n &= (2t)^0\ (2t-1) + (2t)\ (2t-1) + (2t)^2\ (2t-1) + \ldots + (2t)^h\ (2t-1) \\
&= (2t-1)\ (1 + 2t + (2t)^2 + \ldots + (2t)^h\ ) \leftarrow \text{geometric series} \\
&= (2t-1) * [1 - (2t)^{h+1}] / (1 - 2t) \\
&= (2t)^{h+1} - 1
\end{aligned}
$$

$$\Rightarrow \quad h = log_{2t}(n+1) - 1 \quad \Rightarrow \quad h = O(log_{2t}(n+1))$$

Question 3

Now work out the maximum possible height of the B-tree

The tallest B-tree (of which the height is the maximum possible the tree could get) occurs when n elements are arranged in the tree structure such that each node has $(t-1)$ (= the minimum number of) elements. The exception is the root node which can take a single element as the minimum. Let the tree height be h, and consider the number of elements at each height of the tree.

Height/level # of nodes # of elements

0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
.	.	.
.	.	.
.	.	.
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

```
Accordingly, the total number of elements n can be expressed as:
```

$$\begin{aligned} n &= 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \ldots + 2t^{h-1}(t-1) \\ &= 1 + 2(t-1)\ (1 + t + t^2 + \ldots + t^{h-1}) \leftarrow \text{geometric series} \\ &= 1 + [2(t-1) * [1 - t^h] / (1 - t)] \\ &= 2t^h - 1 \end{aligned}$$

$$\Rightarrow \quad h = log_t\left(\tfrac{n+1}{2}\right) \quad \Rightarrow \quad h = O\left(log_t\left(\tfrac{n+1}{2}\right)\right)$$

Question 4

For a branching factor of t= 3, show the resultant B-tree upon inserting the following sequence of letters: {S, Z, G, Y, B, N, D, E, F, U, I, V, M, X, H}.

```
The minimum # of elements for a node (except root) = t - 1 = 2
The maximum # of elements for a node = 2t - 1 = 5
Note:  a node with elements x,y is represented by ||x||y||
Left || represents a pointer to the left child node in the next
level. Right || represents a pointer to the right child node in
the next level.


Initial state:  Null B-tree

Insert S:       S becomes the root
                ||S||

Insert Z:       ||S||Z||

Insert G:       ||G||S||Z||

Insert Y:       ||G||S||Y||Z||

Insert B:       ||B||G||S||Y||Z|| max reached

Insert N:       resolve max node by splitting at the median
                Element and promoting it to the root node. Then
                insert N.
```

```
                    ||S||
         ||B||G||        ||Y||Z||

                       ||S||
         ||B||G||N||        ||Y||Z||

Insert D:
                         ||S||
         ||B||D||G||N||        ||Y||Z||

Insert E:                   ||S||
         ||B||D||E||G||N||        ||Y||Z||
              max reached

Insert F:        Resolve max node first

                ||E||S||
         ||B||D||     ||G||N||     ||Y||Z||

         Now insert F
                    ||E||S||
         ||B||D||    ||F||G||N||     ||Y||Z||

Insert U:
                     ||E||S||
         ||B||D||    ||F||G||N||     ||U||Y||Z||

Insert I:
                      ||E||S||
         ||B||D||    ||F||G||I||N||     ||U||Y||Z||

Insert V:
                      ||E||S||
         ||B||D||    ||F||G||I||N||     ||U||V||Y||Z||
Insert M:
                       ||E||S||
         ||B||D||    ||F||G||I||M||N||     ||U||V||Y||Z||
                         max reached

Insert X:                ||E||S||
         ||B||D||    ||F||G||I||M||N||     ||U||V||X||Y||Z||
                         max reached           max reached

Insert H:        Resolve relevant max node first
```

```
                    ||E||I||S||
||B||D||    ||F||G||   ||M||N||     ||U||V||X||Y||Z||
                                          max reached

Now insert H
                    ||E||I||S||
||B||D||    ||F||G||H||   ||M||N||     ||U||V||X||Y||Z||
                                          max reached
```

Question 5

For deletion the traversal starts from the root. The next node can only be visited if it has more elements than the minimum number of elements. If not, we have to resolve the node.

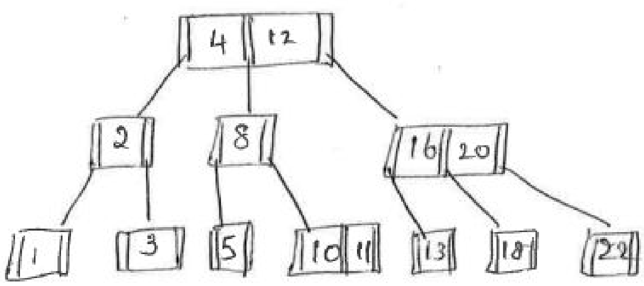
Delete 1

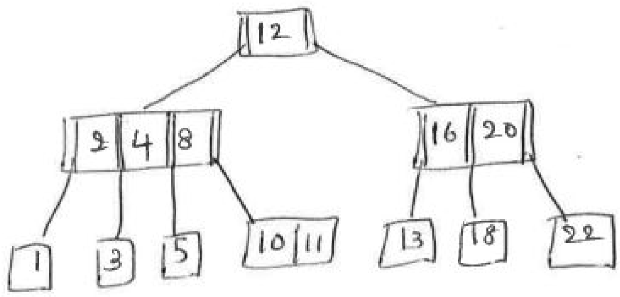
Steps for deleting element 1

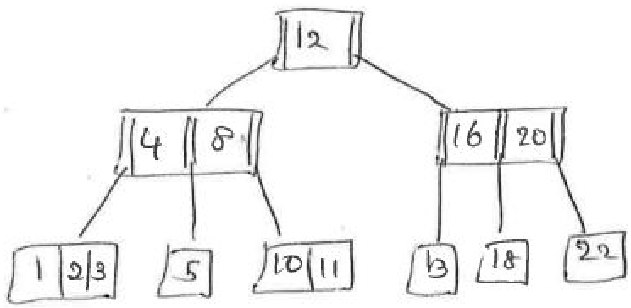
1. Go to root node: [4,12]
2. Check node [2]. It has exactly t-1 elements and its immediate sibling also has exactly t-1 elements → case 3b. Therefore merge parent element 4 with both children: 2 and 8 to form a single node [2,4,8]
   Then goto node [2,4,8].
3. Check node [1]: It has exactly t-1 elements and its immediate sibling also has exactly t-1 elements → case 3b. Therefore merge parent element 2 with both children: 1 and 3 to form a single node: [1,2,3]

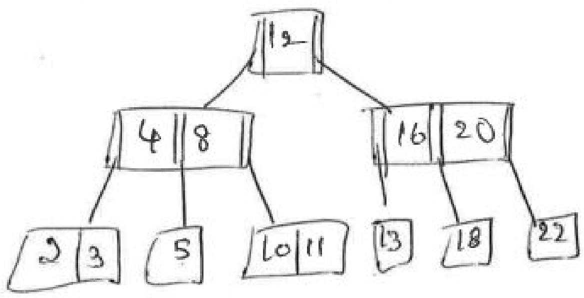
   Then goto node [1,2,3]
4. Now we are at a leaf node with the element to be deleted → case 1 Trivial deletion!

Applying the steps to the current B-tree







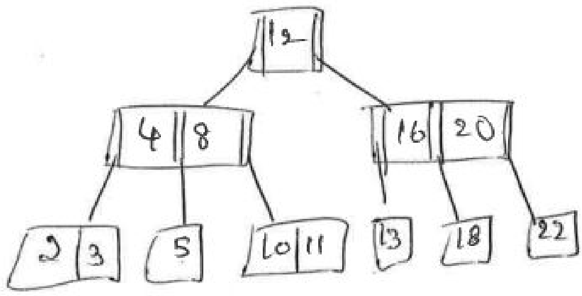


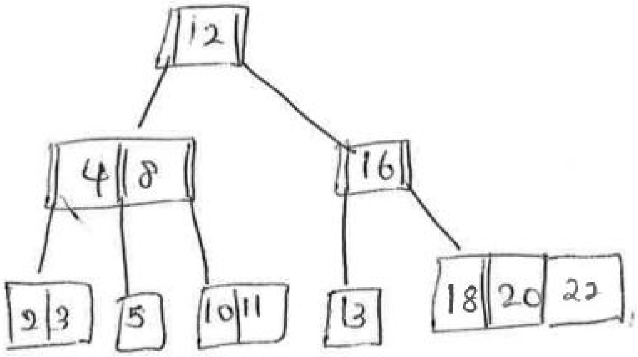
Delete 22

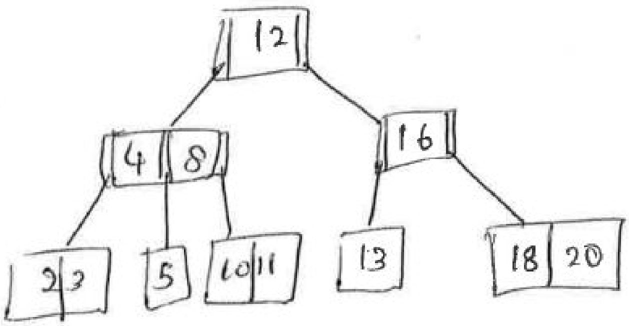
Steps for deleting element 22

1. Go to root node: [12]
2. Check [16,20]. It has at least t elements. Therefore goto node [16,20].
3. Check [22]. It has exactly t-1 elements and its immediate sibling also has exactly t-1 elements → case 3b. Therefore merge parent element 20 with both children: 18 and 22 to form a single node: [18,20,22].
   Then goto [18,20,22]
4. Now we are at a leaf node with the element to be deleted → case 1 Trivial deletion!

Applying the steps to the current B-tree







Delete 16

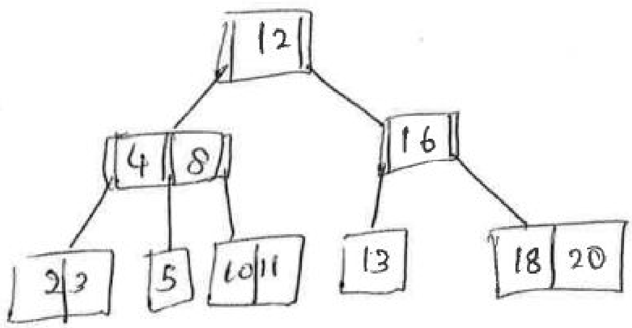
Steps for deleting element 16

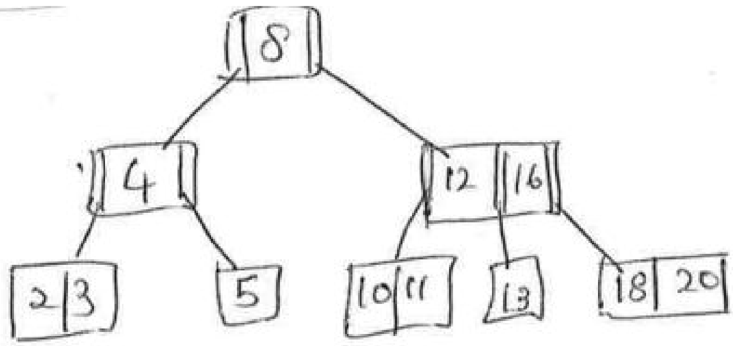
1. Go to root node [12]
2. Check [16]. It has exactly t-1 elements but its immediate sibling [4,8] has at least t elements → case 3a. Therefore we give an extra element to node [16] by rotation => the predecessor element [8] of the sibling node is moved to root node, and element 12 in the root node is moved to node [16]. This results in root node being [8] and the next node to goto being [12,16]. The right child node [10,11] to the element 8

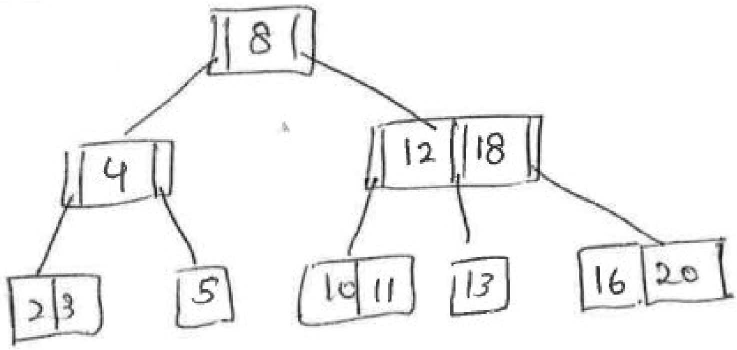
of previous [4,8] node can now be assigned as the left child node to the element 12 of node [12,16].
Then goto node [12,16]

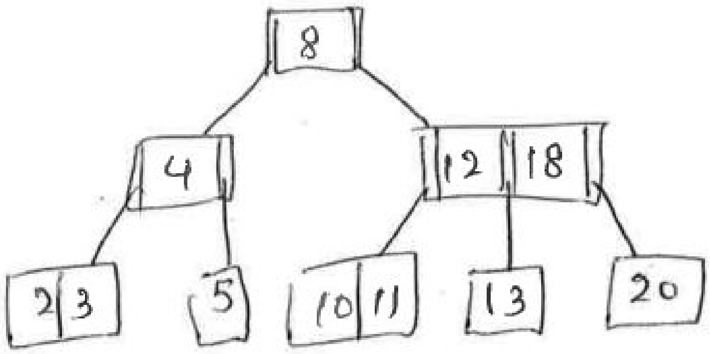
3. Now we are at an internal node with the element to be deleted. With regards to element 16, the left child node [13] has exactly t-1 elements and right child node [18,20] has at least t elements → case 2b. Therefore take the in-order successor of the right subtree: element 18,interchange 16 and 18 so that 16 in [12,16] is replaced by 18, and 18 in [18,20] is replaced by 16.
Then goto [16,20]
4. Now we are at a leaf node with the element to be deleted → case 1 Trivial deletion!

Applying the steps to the current B-tree






8
4
12
18
2
3
5
10
11
13
16
20


8
4
12
18
2
3
5
10
11
13
20