UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 212 Tutorial 6: Version B

- 30/03/2012
- 50minutes
- 3 questions for a total of 30 marks.

Name:______________________________

Student/staff Nr:________________________

Marker (office use):______________________

Question 1 B^*-Trees . . . . (9 marks)

1.1 Describe the problem that arises when the root node in a B^*-tree needs to split. Also suggest (a) solution(s) to this problem. (3)

Answer:

Solution: There are not enough keys to divide between siblings and new root so that siblings remain 2/3 full

Suggestion: Because inserts always happen in leaf nodes, allow new root's children to underflow for now.

1.2 Use your suggestion from the previous question and insert the following keys, in the given order, into an initially empty B^*-tree where $m = 9$. Only draw your final tree. (5)

```
10,11,12,13,14,15,16,17,18,19,20,50,51,52,53,54,55,56,57,58,59,60,40,39,38,37,36,35,34
```

Answer:

1.3 What is the minimum number of keys that may be contained in a B*-tree of order 11 ($m = 11$) and height 3? (1)

Answer:

Solution: Root is 1.

The number of keys in each node is $\lfloor \frac{22-1}{3} \rfloor = 7$

Each node (except the root) has 8 children.

1+ 2*(7) + 2*8*(7) = 127 keys

Question 2 B$^+$-Trees .... (14 marks)

2.1 What is the maximum height that a B$^+$-tree which contains n keys can have? (1)

Answer:

Solution:

The fullness of each leaf is $\lceil \frac{m}{2} \rceil$ and all data are contained in the leaf nodes so there are $\frac{n}{\lceil \frac{m}{2} \rceil}$ leaf nodes.

We need $\frac{\lceil \frac{m}{2} \rceil}{2}$ parents to split these leaves and each parent takes $\lceil \frac{m}{2} \rceil + 1$ children.

The min number of keys is $2(\frac{\lceil \frac{m}{2} \rceil}{2})^{h-1} = n$

solve for h so we get:

$h - 1 = 2 * log_{\frac{\lceil \frac{m}{2} \rceil}{2}} n$

So the height is:

$h = 2 * log_{\frac{\lceil \frac{m}{2} \rceil}{2}} n + 1$

2.2 In terms of searching, is there an advantage of using a standard B-tree over a B$^+$-tree? Motivate your answer. (2)

Answer:

Solution: Data is in the internal nodes as well and can be retrieved quicker.

B+ trees have all data in the leaf nodes so before data can be retrieved traversal proceeds to the lowest level.

2.3 For B-trees m cannot be even. Is this also the case for B$^+$-trees? Motivate your answer. (2)

Solution: Because copies and not keys are sent up and not data one can safely split internal nodes by omitting separators between siblings and dividing children sensibly.

Answer:

2.4 A key in an internal node for a B$^+$ is not always deleted. Why is this and when will it be required that these keys be deleted? (2)

Solution: They are still good guides for searching and deletion may become expensive.

They are deleted as soon as their children merge because of a delete.

Answer:

2.5 Discuss one B$^+$-tree issue that is addressed by prefix B$^+$-trees. (2)

Solution: Separators need only be prefixes, allows for more space in internal nodes meaning more children meaning shallower trees.

Answer:

2.6 Insert the following keys, in the given order, into a `prefix B`$^+$-tree of order 5. Only draw your final tree. (5)

`22222, 33345, 45678, 110866, 110877, 110856, 33444, 444098, 120899, 321123.`

Answer:

Question 3 Tries ....(7 marks)

3.1 What is the height of a trie containing only the following words? Also mention how you determined your answer. (2)

`need, needs, no, exception, excuse, excused, ether, man, mouse, electroencepphalographically, electricity.`

Answer:

Solution: 8

3.2 Apply the `compressTrie` algorithm to the trie from the previous question and draw the compressed version on this trie. (5)

Answer: