Homework 4

IMPORTANT

- **Plagiarism**: Homework may be discussed with other students, but all homework is to be completed **individually**.
- **Typeset** all of your answers whenever possible. Illegible handwriting may get no points, at the discretion of the graders.
- **Deposit hard copy** of your answers in **class at 1:30pm on Thursday, February 27th**. For ease of grading, please
  - solve each of the 5 questions on a **separate** page. If you need more pages for one question, please staple them together.
  - Type the full info on each page: your **name**, **Andrew ID**, **course #** , **Homework #** , **Question #** on each of the 5 pages.
- **Late homeworks**: If you are turning your homework in late, please email it
  - to all TAs
  - with the subject line exactly `15-415 Homework Submission (HW 4)`
  - and the count of slip-days you are using.

For your information:

- Graded out of **100** points; **5** questions total
- Rough time estimate: $\approx$ 2-4 hours

Revision : 2014/03/09 15:17

Question	Points	Score
ISAM Tree	10	
B+ Tree Bulk-Loading	15	
B+ Trees	20	
Extendible Hashing and Linear Hashing	30	
External Sorting	25	
Total:	100	

Question 1: ISAM Tree..............................[10 points]
Submit on separate page

For the following sub-questions, consider the ISAM tree structure with order of d = 1 (i.e. there are at most 2 keys per node, and at most 3 pointers to children).
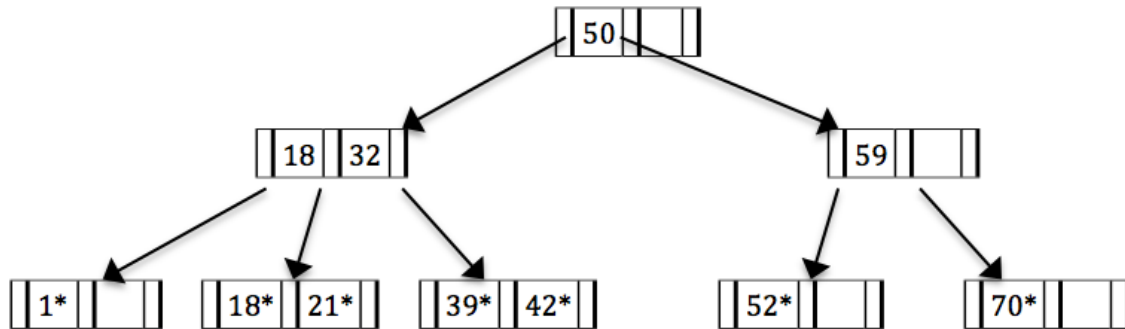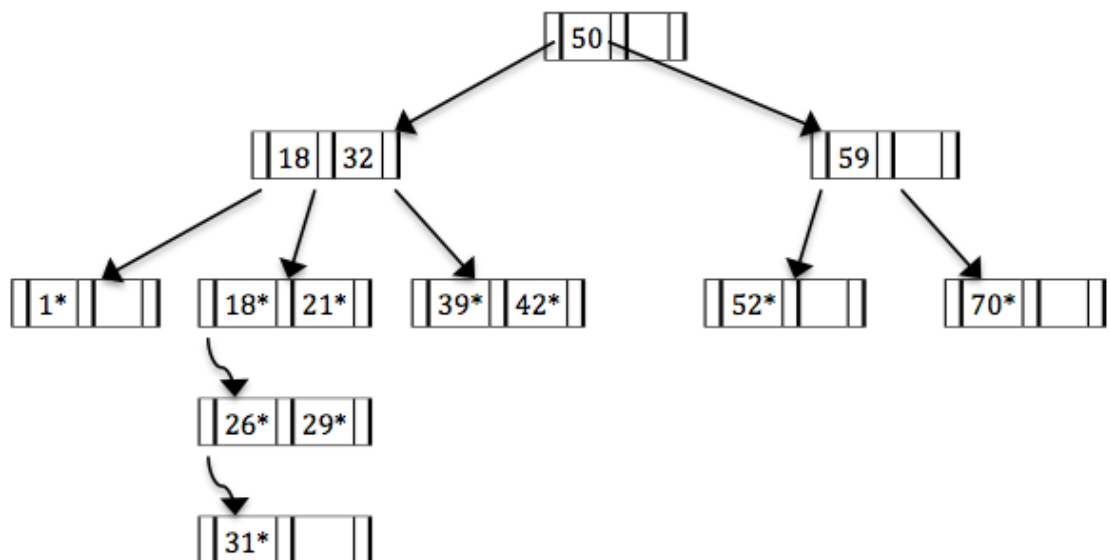

Figure 1: ISAM tree structure for Question 1

(a) [**10 points**] show the new structure after inserting keys 26, 29 and 31.

Solution:

(a)

Grading info: *Graded by Vagelis Papalexakis.*

- *-1 point if a node is missing a key or value.*

- *-0.5 point if the arrows for the overflow pages are missing.*

- *-2 for every node that has more than 2 keys.*

Question 2: B+ Tree Bulk-Loading .... [15 points]
Submit on separate page

Consider an empty B+ Tree with order $d = 2$ (i.e. there are at most 4 keys per node, and at most 5 pointers to children). Bulk load the B+ tree with data entries with integers from 1 to 70 (i.e. 1,2,3,...,70) so that each leaf is full, using the algorithm outlined in Section 10.8.2 (Page 360) of the textbook.

(a) [**5 points**] What is the height of the resulting tree after inserting all the above keys? (i.e. The height of the tree in Figure 2 is three)

(b) [**5 points**] Using the bulk-load algorithm described in Section 10.8.2, what's the maximum number of entries that can be inserted to the tree while keeping its height to be three?

(c) [**5 points**] Print out the content in the root node after inserting 1*, 2*, ..., 70*.

Solution:

(a) The height of the resulting tree is 4

(b) 68

(c) Only "37" in the root node

Grading Info:

- -7 if the answer does not follow algorithm but creates a sort of valid tree

- -2 for an answer of 64 for part (b)

- -2 for an answer of 36 for part (c)

- -3 for an answer of 100 for part (b)

- -4 for an answer of 3 for part (a)

- -4 for part (b) if some logic is shown but it is very wrong

Question 3: B+ Trees .... [20 points]
Submit on separate page

For the following sub-questions, consider the B+ tree structure with order of $d = 1$ (i.e. there are at most 2 keys per node, and at most 3 pointers to children).
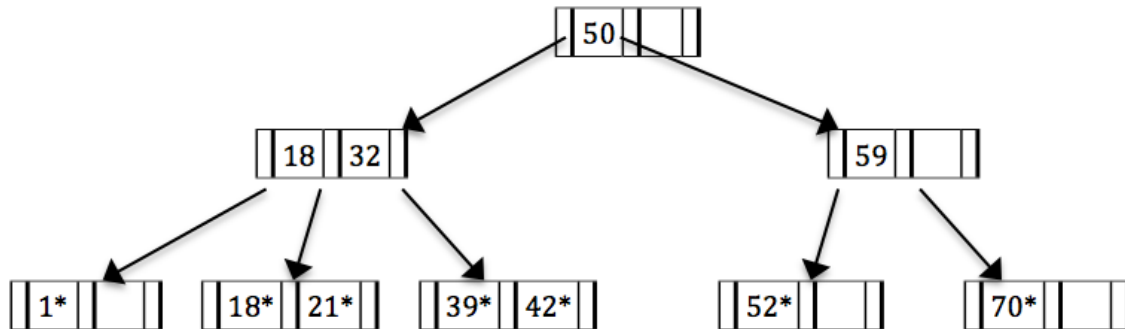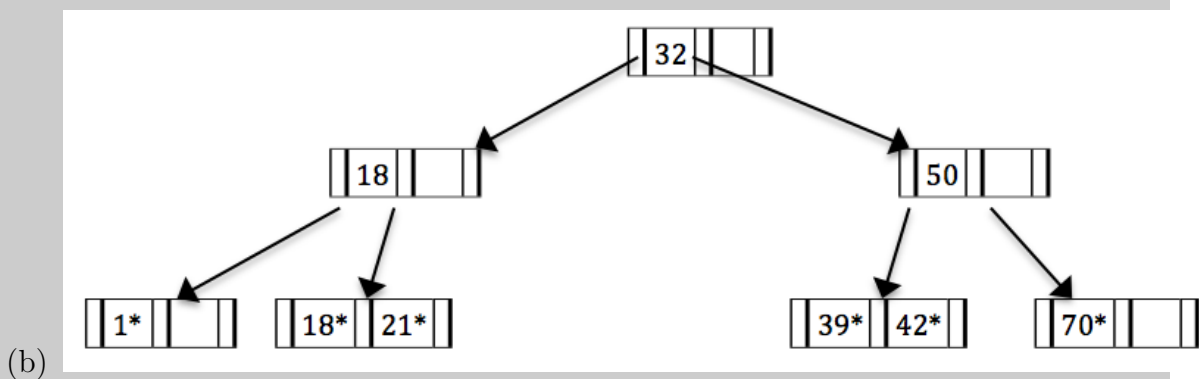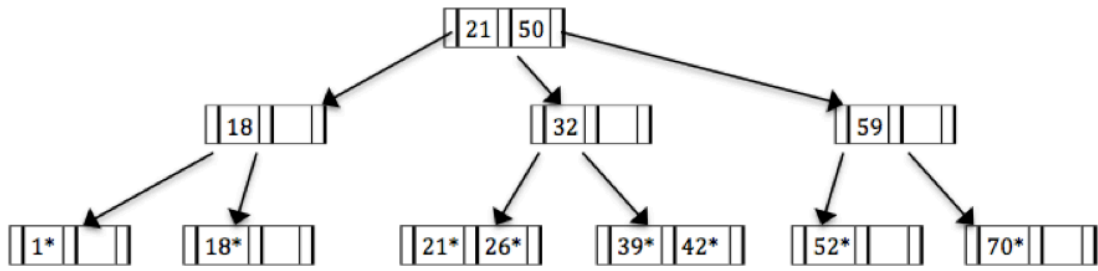

Figure 2: B+ tree structure for Question 3

(a) Starting from the original B+ tree in Figure 2, we insert the record 26* with a key 26.

  i. [5 points]   From the choices in Figure 3, which is the resulting tree if we use the default "no redistribution" strategy: A, B, C, D, or neither? If neither, draw the correct tree.

  ii. [5 points]   From the choices in Figure 3, which is the resulting tree if we use the "redistribution with left sibling" strategy: A, B, C, D, or neither? If neither, draw the correct tree.

(b) [10 points]   Starting from the original B+ tree in Figure 2, show the resulting tree after deleting record 52 if we can borrow node from the left sibling.
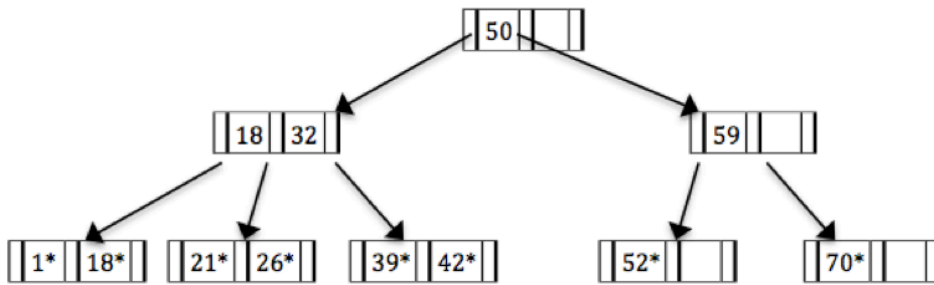
Solution:

(a)   1. A
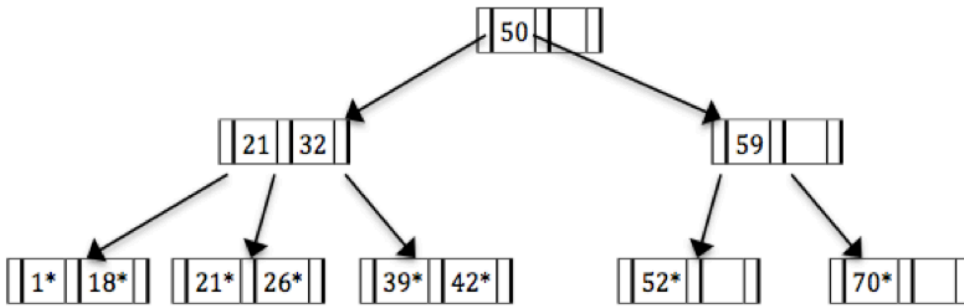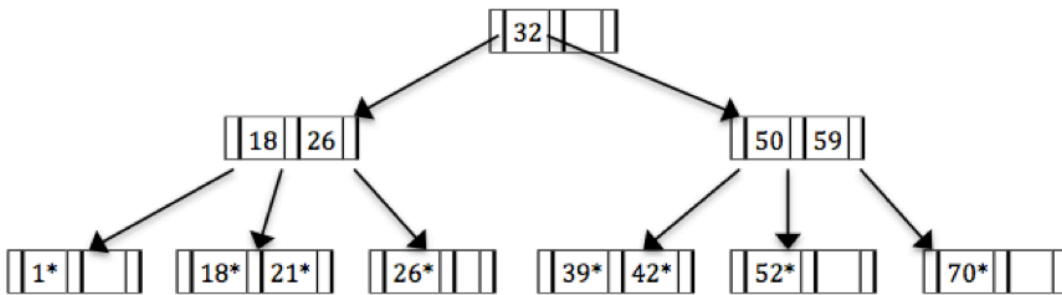
      2. C

(b)

Tree A

Tree B

Tree C

Tree D

Figure 3: Figure for part (a)

Grading info:

- _-3: wrong tree structure_

- _-2: breaking B+ tree rules_

- _-1: Any wrong keys in the tree_

- _-0.5: No * in the leaf node_

Question 4: Extendible Hashing and Linear Hashing . [30 points]
Submit on separate page

Assume we have the following records where we indicate the hashed key in parenthesis (in binary)

```
a [11001]
b [00111]
c [00101]
d [00110]
e [10101]
f [01000]
g [00011]
h [11110]
i [00001]
```

(a) Consider an Extendible Hashing structure where buckets can hold up to three records. Initially the structure is empty (only one empty bucket). Consider the result after the records above have been inserted in the order shown, using the highest-bits for the hash function. The directory doubles in size after **very few** overflows.

   i. [**5 points**]   What will be the global depth of the resulting directory?

   ii. [**5 points**]   How many buckets will we have?

   iii. [**5 points**]   List all the elements in the bucket which contains the element "a" (including "a"). What is the local depth of this bucket?

> **Solution:**
>
>   i. 3
>
>   ii. 4
>
>   iv. bucket contains a,e,h; The local depth is 1.
>
> *Grading info:*
>
> - *-1: missing depth in part (iii)*

(b) For the same records above, consider a linear hashing structure where buckets can hold up to three records. Initially the structure is empty (only one empty bucket). Show the linear hashing structure after these records (in the order shown above) have been inserted. Assume that we split whenever the new key goes into a full bucket (which may or may not be the split bucket).

   i. [**5 points**]   How many buckets will we have (overflow pages do not count)?

ii. [**5 points**] List all the elements in the bucket which contains the element "d" (including "d"). If there is overflow page for the bucket, also list the elements in the overflow page(s).

iii. [**5 points**] List all the elements in the bucket which contains the element "a" (including "a"). If there is overflow page for the bucket, also list the elements in the overflow page(s).

Solution:

i. 5

ii. d,h; There is no overflow page.

iii. a,c,e,i; i is in the overflow page.

Grading info:

- *-1: missing overflow page*

- *partially correct answer will get points accordingly*

Question 5: External Sorting .... [25 points]
Submit on separate page

Suppose you have a file with $N = 5 \times 10^7$ pages.

(a) [**10 points**] What is the total I/O cost of sorting the file using the two-way merge sort (with three buffer pages)?

> **Solution:** Using the analysis and formulas on chapter 13.2 (p.423), we can get the result to be 2.7×10^9 I/O.

(b) Now suppose we have 129 buffer pages and we are using the external sorting algorithm discussed in chapter 13.3 (p.424).
  - i. [**5 points**] How many runs will you produce in the first pass?
  - ii. [**5 points**] Now assume that we have a disk with an average seek time of 10ms, average rotation delay of 5ms and a transfer time of 1ms for each page. Assuming the cost of reading/writing a page is the sum of those values(i.e. 16ms) and do not distinguish between sequential and random disk-access - **any** access is 16ms, what is the total running time to sort the file?
  - iii. [**5 points**] With 129 buffer pages, what's the maximum size of the file (in number of pages) that we can sort with 3 passes?

> **Solution:**
>
>   i. 387597 runs.
>
>   ii. $6.4 \times 10^9 ms$
>
>   iii. $128 * 128 * 129 = 2113536$ pages
>
> _Grading info:_
>
>   - _-1: missing unit_
>   - _-1: close numeric errors_
>   - _large error will not be given partial credit if no explanation is given_