

CARNEGIE MELLON UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
15-445/645 – DATABASE SYSTEMS (FALL 2018)
PROF. ANDY PAVLO

Homework 2 (by Ao Zeng) – Solutions
Due: **Friday Sept 28, 2018 @ 11:59pm**

IMPORTANT:

- **Upload this PDF** with your answers to **Gradescope by 11:59pm on Friday Sept 28, 2018.**
- **Plagiarism:** Homework may be discussed with other students, but all homework is to be completed **individually**.

For your information:

- Graded out of **100** points; **4** questions total
- Rough time estimate: \approx 1-4 hours (0.5-1 hours for each question)

Revision : 2018/10/10 19:25

Question	Points	Score
Extendible Hashing	30	
Linear Hashing	20	
B+Tree	40	
Skip List and Radix Tree	10	
Total:	100	

Question 1: Extendible Hashing.....[30 points]**Graded by:**

(a) Consider an extendible hashing structure that

1. Each bucket can hold up to 2 records
2. Is initially empty (only one empty bucket)

Consider the result after inserting key 8, 16, 4, 3, 11, 12 in order, using the lowest-bits for the hash function. That is, records in a bucket of local depth d agree on their **rightmost** d bits. For example, key 4 (0100) and key 12 (1100) agree on their rightmost 3 bits (100).

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

☐ 0 ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

ii. **[5 points]** Now insert key 18. What is the local depth of the bucket that contains the key 18?

☐ 0 ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

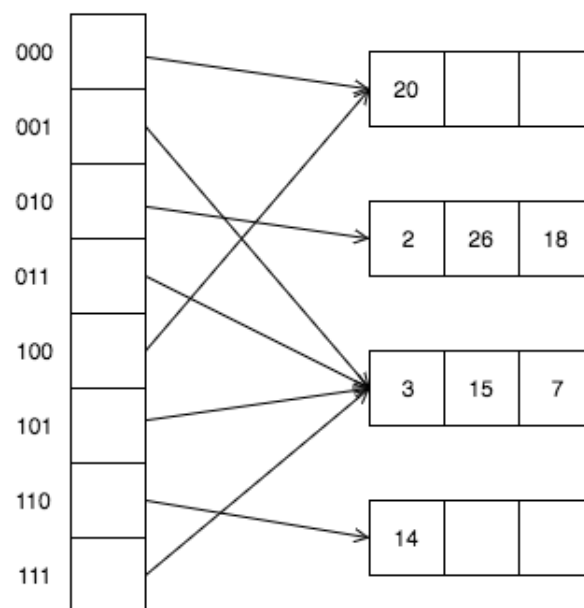


Figure 1: Extendible hashing

(b) Answer the following questions about Figure 1. Suppose we insert keys 28, 30, 4, 8, 34 in order.

i. **[5 points]** Which key will cause the first split?

☐ 28 ☐ 30 ☐ 4 ☒ 8 ☐ 34 ☐ None of the above

ii. **[5 points]** Which key will first cause the directory to double in size?

☐ 28 ☐ 30 ☐ 4 ☐ 8 ☒ 34 ☐ None of the above

- (c) Start with the original hash table shown in Figure 1. Consider the result after deleting keys 14, 20 in order. (Try to coalesce when an existing bucket becomes empty).
- i. **[5 points]** How many buckets will remain?
☐ 0 ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ None of the above
- ii. **[5 points]** What is the local depth of the bucket containing key 2?
☐ 0 ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ None of the above

Question 2: Linear Hashing.....[20 points]**Graded by:**

Answer the following questions for the hash table of Figure 2. Assume that a bucket split occurs whenever an overflow page is created. $h_0(x)$ takes the rightmost 2 bits of key x as the hash value, and $h_1(x)$ takes the rightmost 3 bits of key x as the hash value.

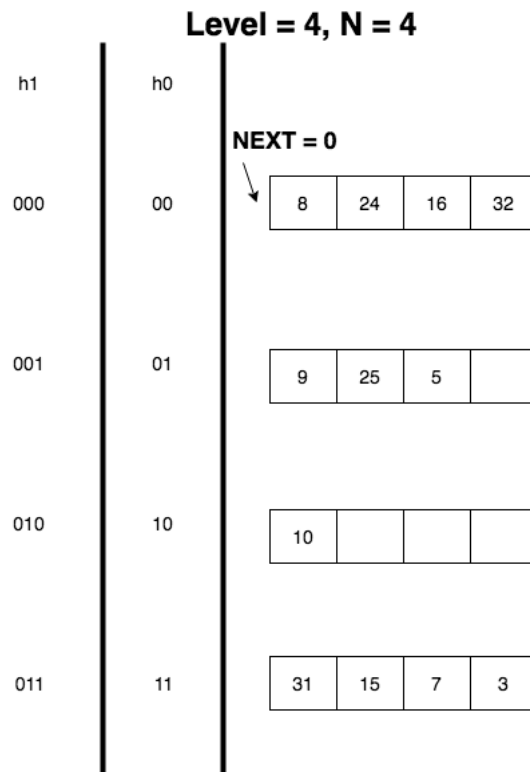


Figure 2: Linear Hashing

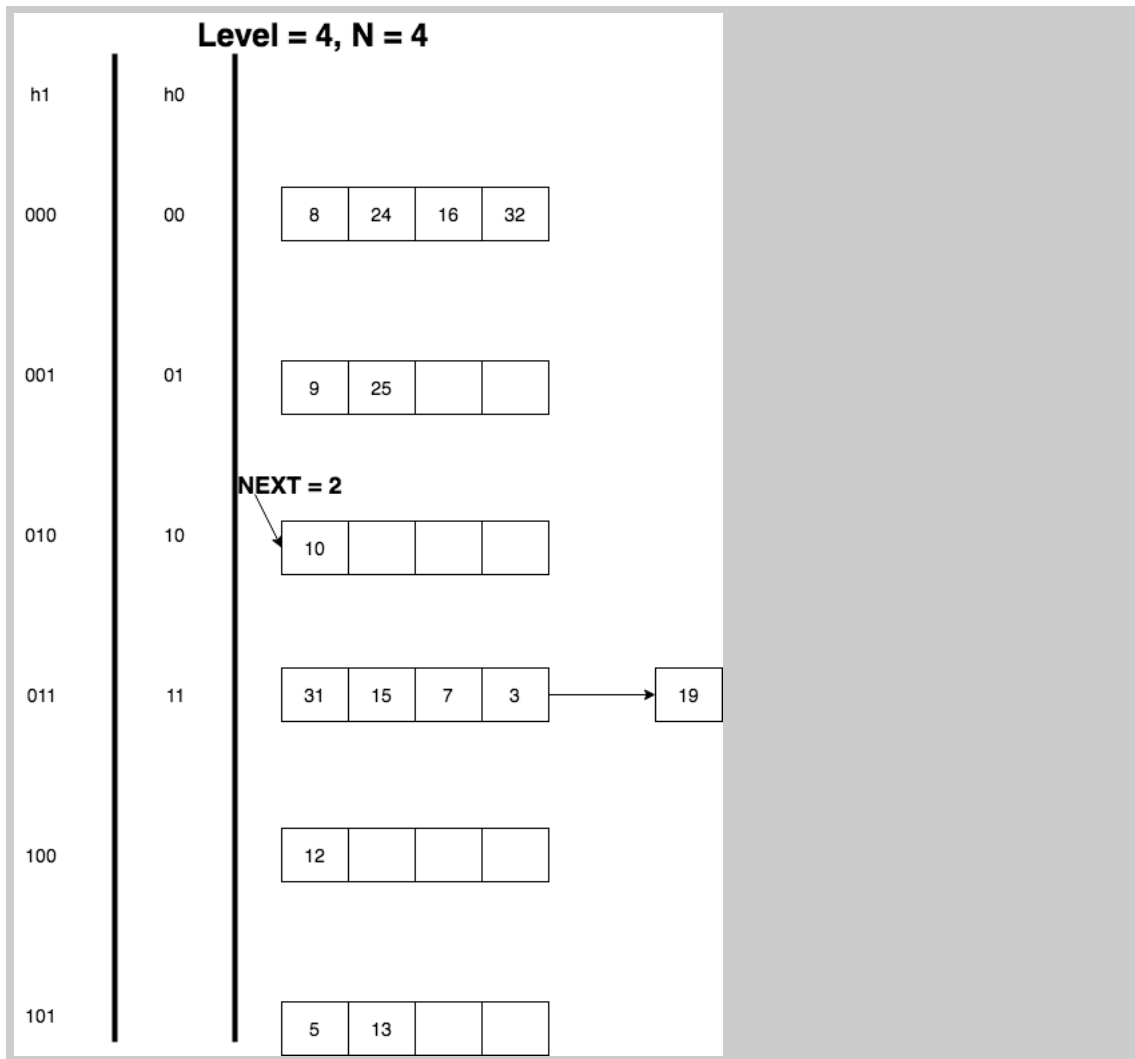
(a) **[5 points]** What is the smallest key that is larger than 25 whose insertion will cause a split? ☐ 26 ☒ 27 ☐ 28 ☐ 29 ☐ 30 ☐ None of the above

(b) **[15 points]** Starting from the hash table of Figure 2, plot the final hash table, after inserting 12, 13, 19. Remember to indicate the new hash function (if any), and to update the “Next” pointer, if needed.

Use the following draw.io template for your answer:

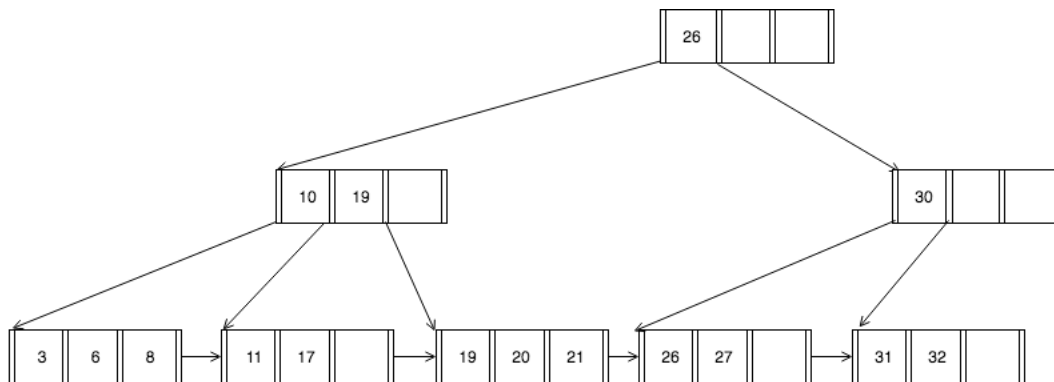
<https://cmudb.io/fall2018-hw2-q2>

Solution: See below



Question 3: B+Tree.....[40 points]**Graded by:**

(a) [5 points] Consider the B+tree shown in Figure 3.

Figure 3: B+ Tree of order $d = 4$ and height $h = 3$.

What is the minimum number of pointers to be followed to satisfy the query: Get all records with key greater than 11 and less than 27?

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5 ☐ None of the above

(b) Using the B+tree in Figure 3 of order $d = 4$ and height $h = 3$ levels, make the following assumptions:

- With respect to “ \geq ”, follow the convention used in the textbook, and in Figure 3, that is, the left pointer is for $<$, the right one for \geq .
- In case of underflow, if you can borrow from both siblings, choose the one on the *right*.

For all questions below, use the standard B+tree algorithm given in the slides and the textbook (on insertions: 2-to-1 split, no deferred splits; on deletions: no underflowing pages).

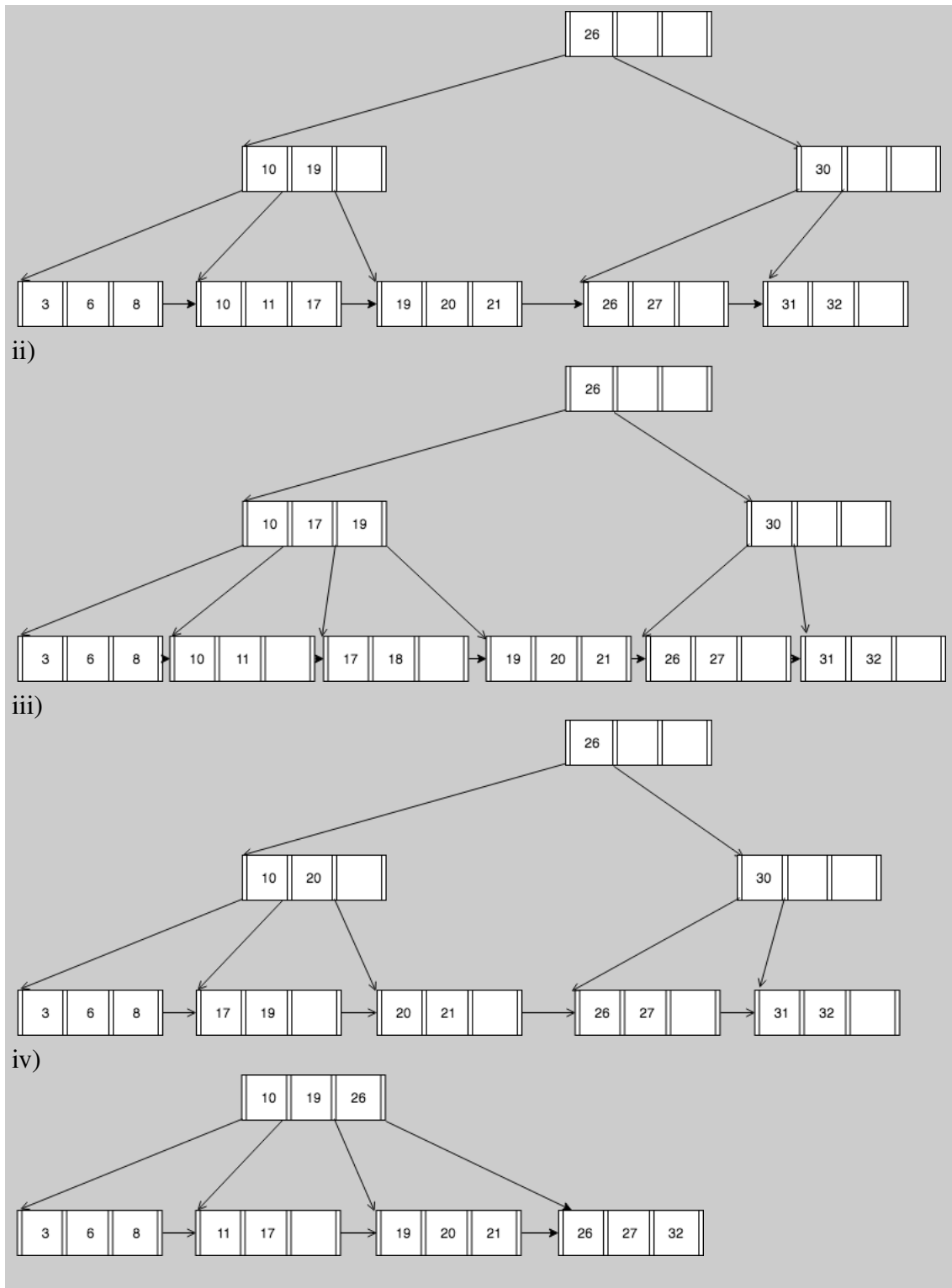
Use the following draw.io template for your answer:

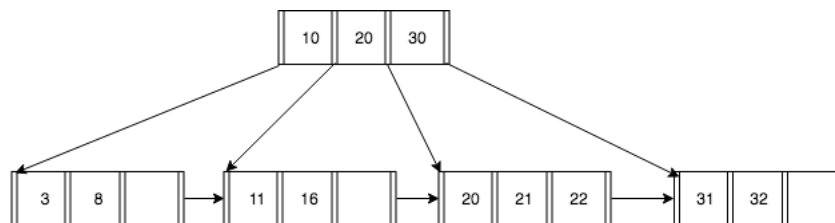
<https://cmudb.io/fall2018-hw2-q3-1>

NOTE: In all cases, start from the B+tree of Figure 3.

- [5 points] Start from the original B+tree; insert 10^* .
- [6 points] Start from the original B+tree; insert 10^* , 18^* .
- [6 points] Start from the original B+tree; delete 11^* .
- [6 points] Start from the original B+tree; delete 31^* .

Solution: i)



Figure 4: B+ Tree of order $d = 4$ and height $h = 2$.

- (c) Consider the B+tree in Figure 4 of order $d = 4$ and height $h = 2$ levels. Please make the same assumptions that are in the previous question. For all questions below, use the standard B+tree algorithm given in the slides and the textbook (on insertions: 2-to-1 split, no deferred splits; on deletions: no underflowing pages).

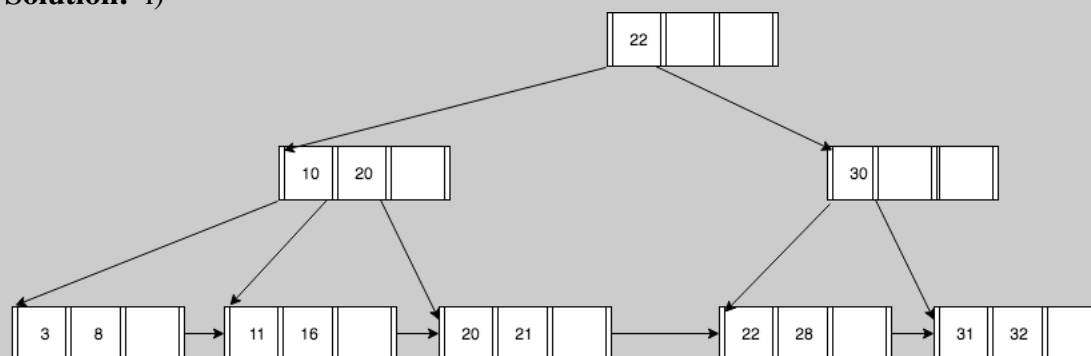
Use the following draw.io template for your answer:

<https://cmudb.io/fall12018-hw2-q3-2>

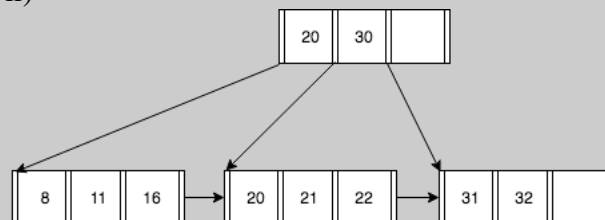
NOTE: In all cases, start from the B+tree of Figure 4.

- i. [6 points] Start from the original B+tree; insert 28^* .
- ii. [6 points] Start from the original B+tree; delete 3^* .

Solution: i)



ii)



Question 4: Skip List and Radix Tree.....[10 points]

Graded by:

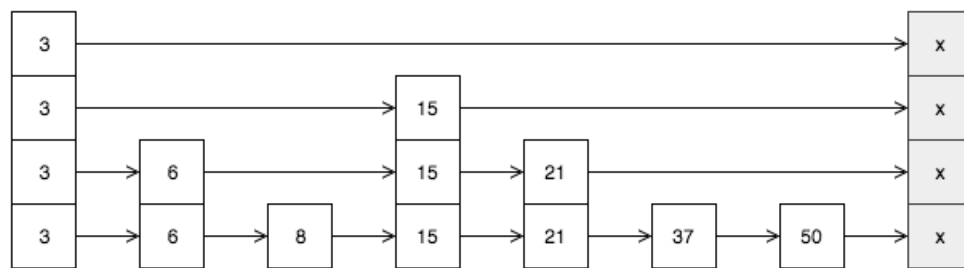


Figure 5: Skip List

- (a) [5 points] Consider the skip list in Figure 5. Suppose we want to insert key 20. Which of the following node(s) might directly point to the new node? Select all that apply.

☒ 3 ☐ 6 ☐ 8 ☒ 15 ☐ 21 ☐ 37 ☐ 50 ☐ Nil ☐ None of the above

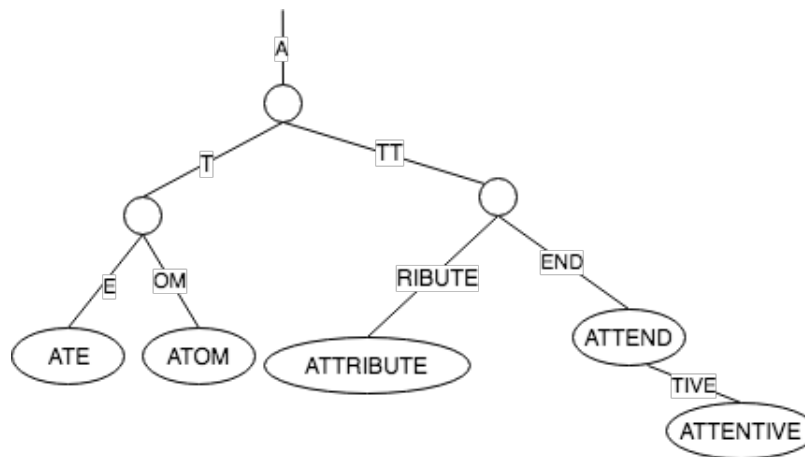


Figure 6: Radix Tree

- (b) [5 points] Consider the radix tree in Figure 6. Is it a valid radix tree? If yes, draw the tree after inserting the new word “APPROVE”. Otherwise, draw a valid radix tree with existing words.

Use the following draw.io template for your answer:

<https://cmudb.io/fall2018-hw2-q4>

Solution: Not valid

