CS3223 Tutorial 1

27 Jan 2022 (Courtesy of Sean)

Question 1

1. Consider the Megatron 747 disk with the following characteristics:
   - There are four platters providing eight surfaces, each surface has $2^{13} = 8192$ tracks, each track has $2^8 = 256$ sectors, and each sector has $2^9 = 512$ bytes.
   - The disk rotates at 3840 rpm.
   - To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
   - 10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

   Answer the following questions.
   a) What is the capacity of the disk?
   b) What is the minimum and maximum time it takes to read a 4096-byte block (8 consecutive sectors) from the disk?
   c) Suppose that we know that the last I/O request accessed cylinder 1000. What is the expected (average) block access time for the next I/O request on the disk?

Question 1(a)

a) What is the capacity of the disk?

There are four platters providing eight surfaces, each surface has $2^{13} = 8192$ tracks, each track has $2^8 = 256$ sectors, and each sector has $2^9 = 512$ bytes.

Question 1(a)

a) What is the capacity of the disk?

There are four platters providing eight surfaces, each surface has $2^{13} = 8192$ tracks, each track has $2^8 = 256$ sectors, and each sector has $2^9 = 512$ bytes.

1 Sector = 2^9 bytes

1 Track = $2^9 * 2^8 = 2^{17}$ bytes

1 Surface = $2^{17} * 2^{13} = 2^{30}$ bytes

1 Disk = 4 Platters = $2^{30} * 8 = 2^{33}$ bytes

Question 1(b)

b) What is the minimum and maximum time it takes to read a 4096-byte block (8 consecutive sectors) from the disk?

The disk rotates at 3840 rpm.
To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

Question 1(b)

b) What is the minimum and maximum time it takes to read a 4096-byte block (8 consecutive sectors) from the disk?

The disk rotates at 3840 rpm.
To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

Rotational Time = 60 / 3840 = 0.015625s = 15.625ms

Rotational Time for Sector + Gap = 15.625 / 256 = 0.061ms

Rotational Time for Sector (W/O Gap) = 0.061 * 0.9 = 0.055ms

Question 1(b)

b) What is the minimum and maximum time it takes to read a 4096-byte block (8 consecutive sectors) from the disk?

The disk rotates at 3840 rpm.
To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

Transfer Time to transfer 8 sectors + 7 gaps =
7 * 0.061 + 0.055 = 0.482ms

Min Time = 0ms + 0ms + 0.482ms = 0.482ms

Max Time = 17.4ms + 15.625ms + 0.482ms = 33.507ms

(Calculated using Seek Time + Rotational Delay + Transfer Time)

Question 1(c)

c) Suppose that we know that the last I/O request accessed cylinder 1000. What is the expected (average) block access time for the next I/O request on the disk?

The disk rotates at 3840 rpm.
To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

Question 1(c)

c) Suppose that we know that the last I/O request accessed cylinder 1000. What is the expected (average) block access time for the next I/O request on the disk?

The disk rotates at 3840 rpm.
To move the head assembly between cylinders takes 1 ms to start and stop, plus 1 ms for every 500 cylinders traveled. Thus, the heads move one track in 1.002 ms, and move from the innermost to the outermost track, a distance of 8192 tracks, in about 17.4 ms.
10% of the space (on top of the usable space) are used for gaps, i.e., between 2 sectors, there is a gap that is not used to store data.

Average number of cylinders travelled =
(999+998+...+1+0)/8192 +(1+...+7192)/8192 = 3218.45

Average Seek Time = 1 + 1 * (3218.45/500) = 7.44ms

Average Rotational Delay = 15.625/2 = 7.8125ms

Total I/O TIme = 7.44 + 7.8125 + 0.482 = 15.73ms

Note: On average, it takes about half a rotation for the correct sector to come under the head.

Question 1

Learning Objective:

1. Understand how disk storage works on a simplified level
2. Understand the concepts of seek time, rotational delay and transfer time and how they play a part in I/O cost

Question 2

2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.

Question 2

2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.

10

Insert: 10

Question 2

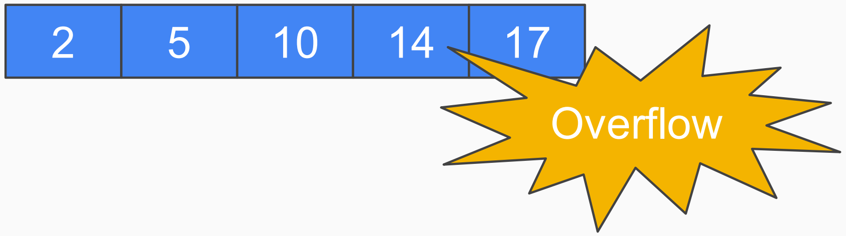
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.

2	5	10	14

Insert: 2, 5, 14

Question 2

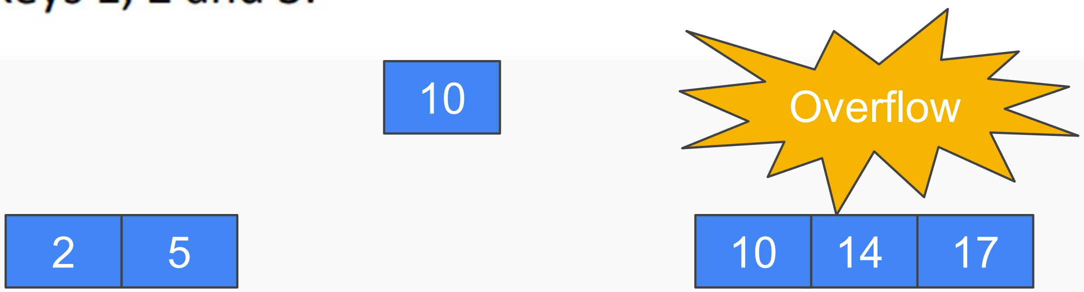
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 2, 5, 14, 17

Question 2

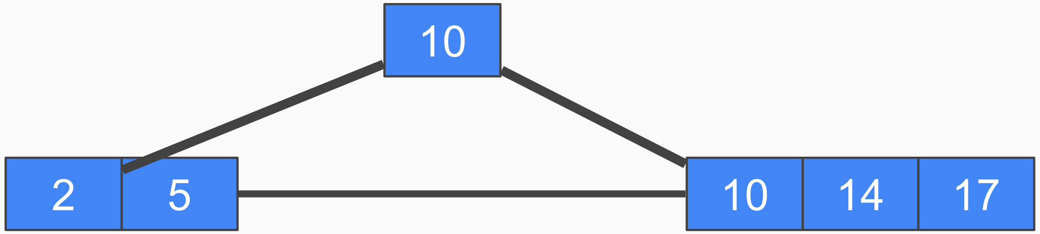
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 2, 5, 14, 17

Question 2

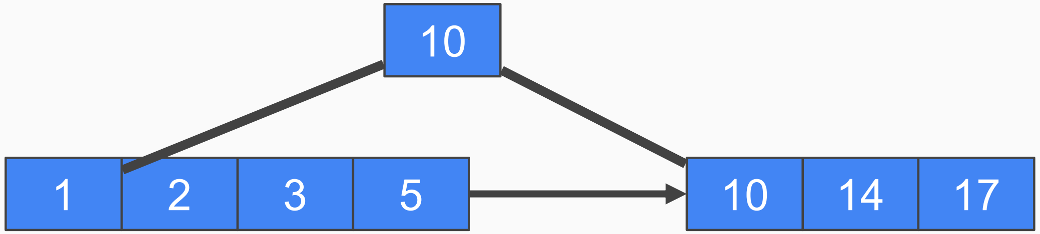
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 2, 5, 14, 17

Question 2

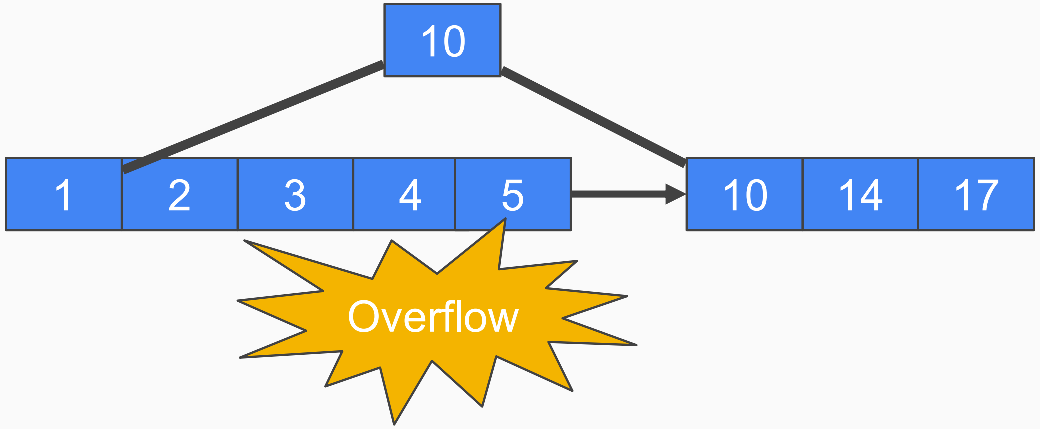
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3

Question 2

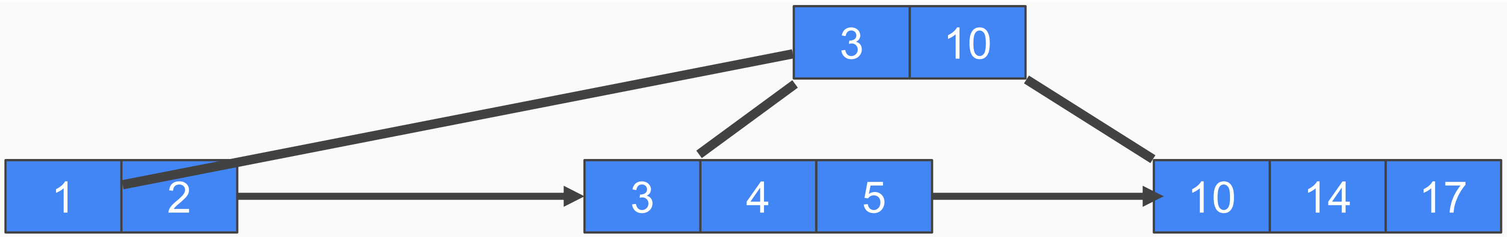
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Insert: 1, 3, 4

Question 2

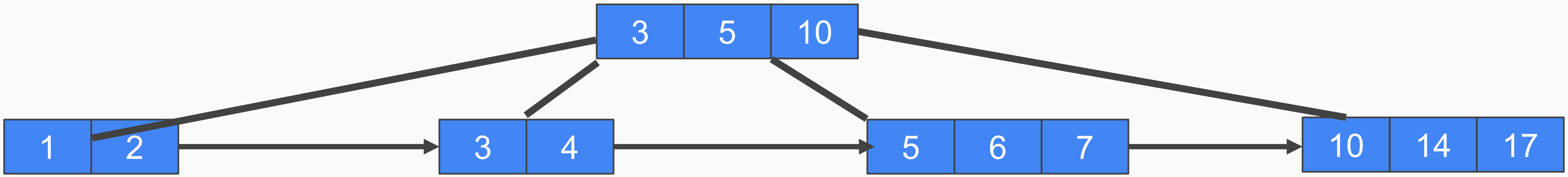
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4

Question 2

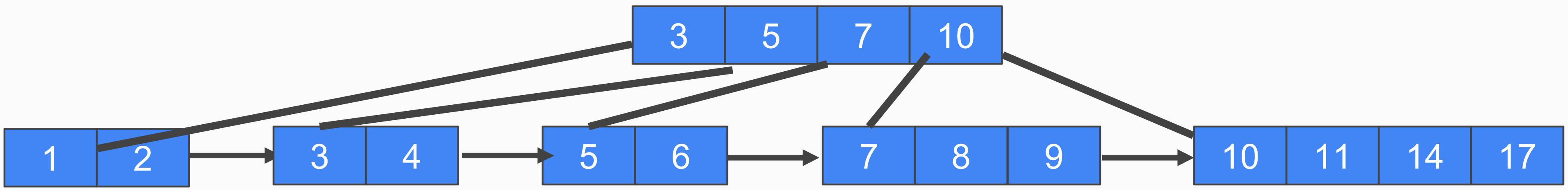
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7

Question 2

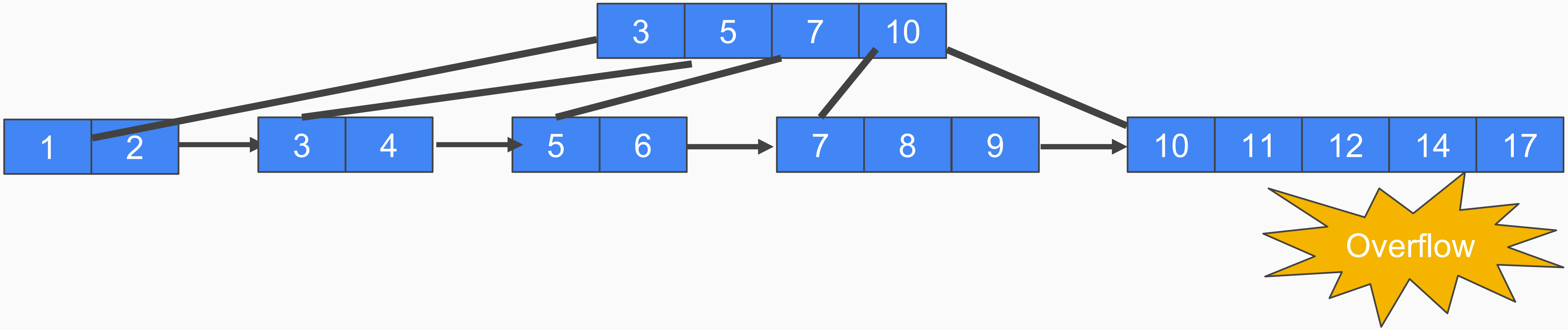
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11

Question 2

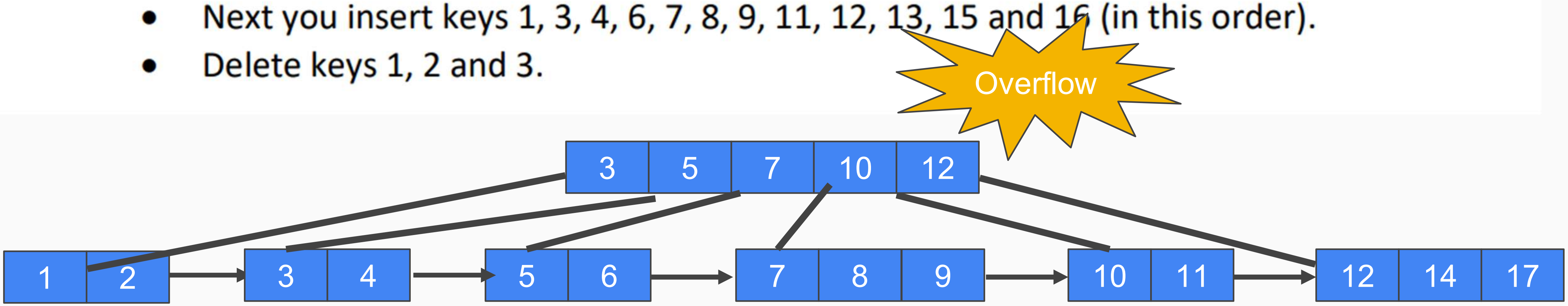
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11, 12

Question 2

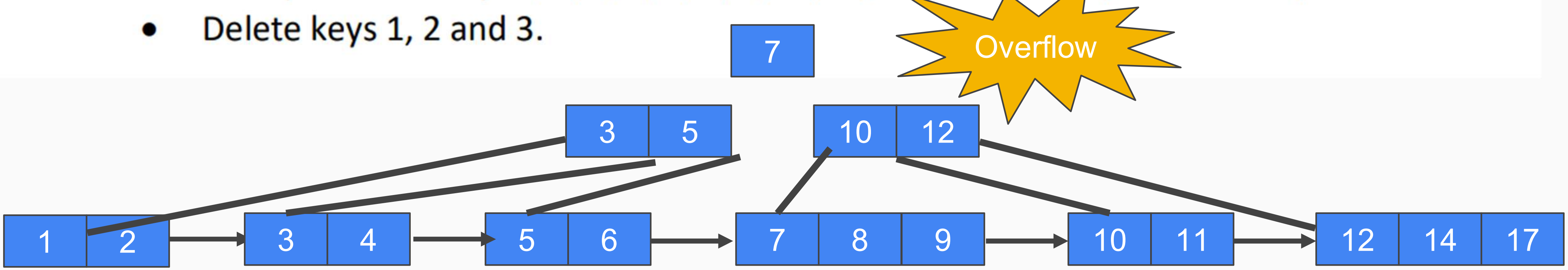
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11, 12

Question 2

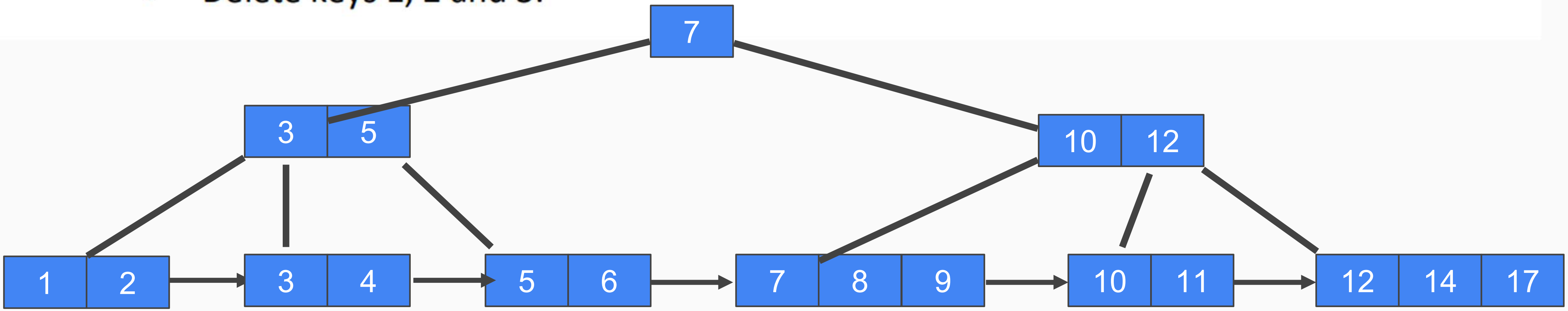
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11, 12

Question 2

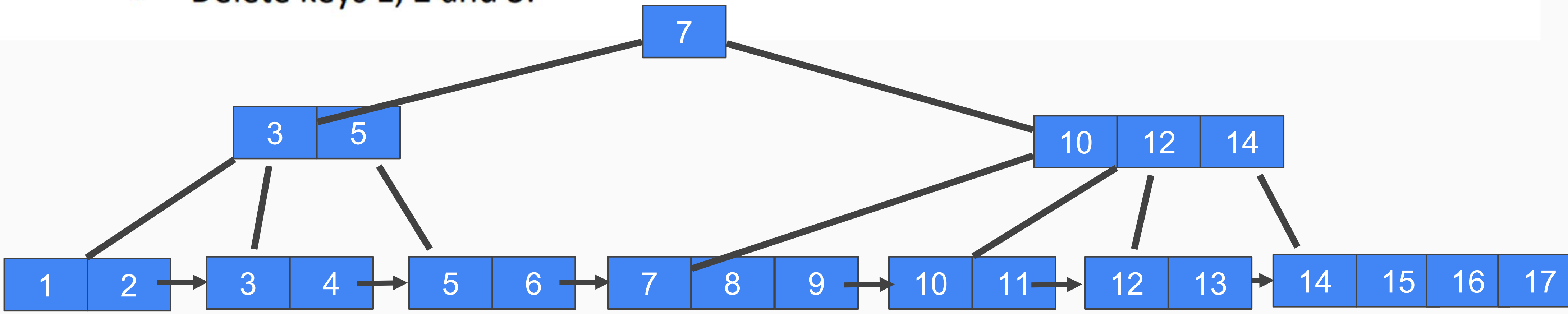
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11, 12

Question 2

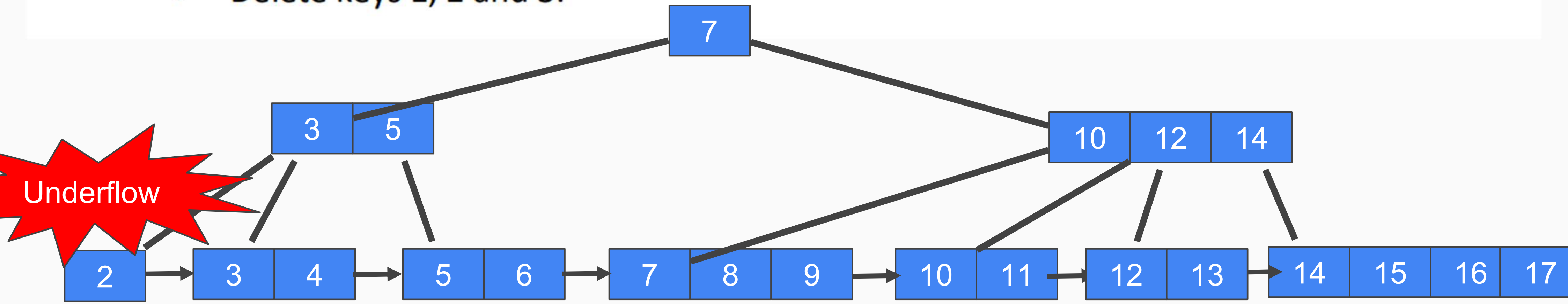
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Insert: 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15, 16

Question 2

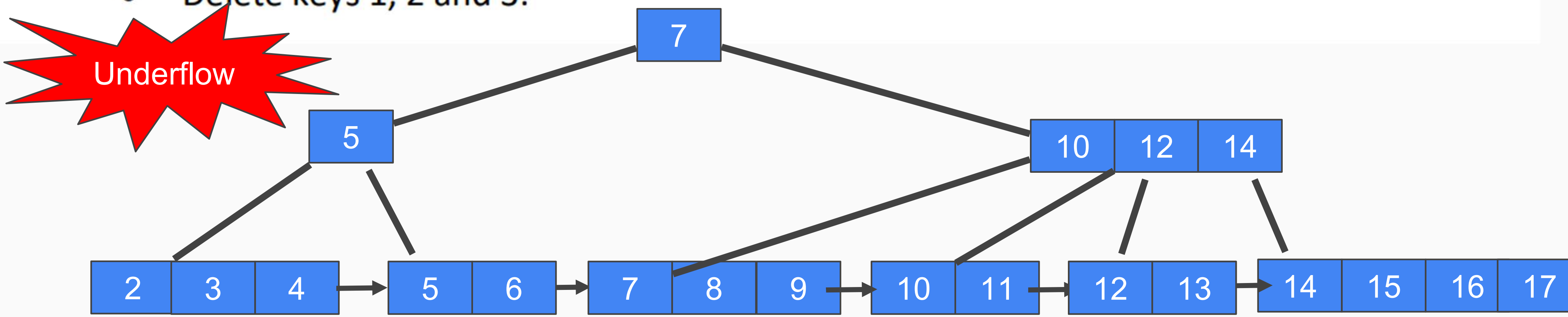
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Delete: 1

Question 2

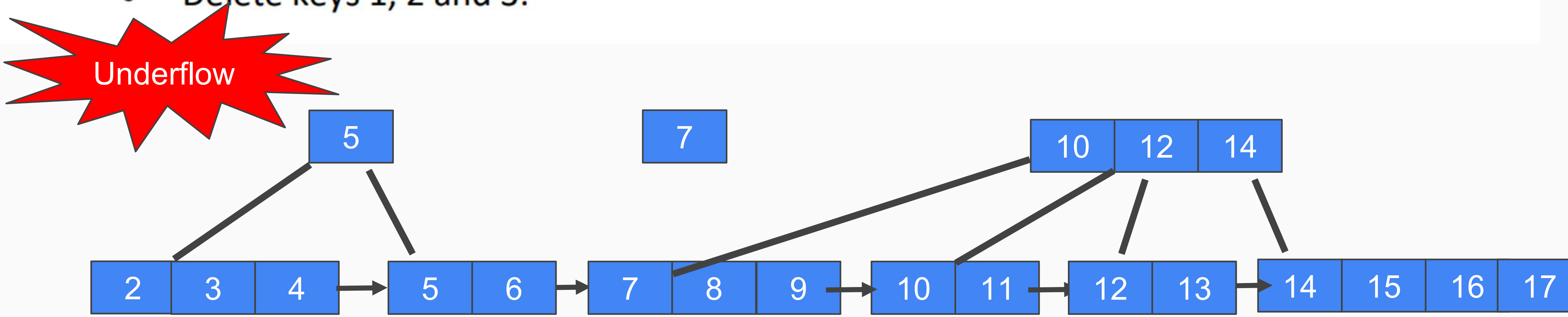
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Delete: 1

Question 2

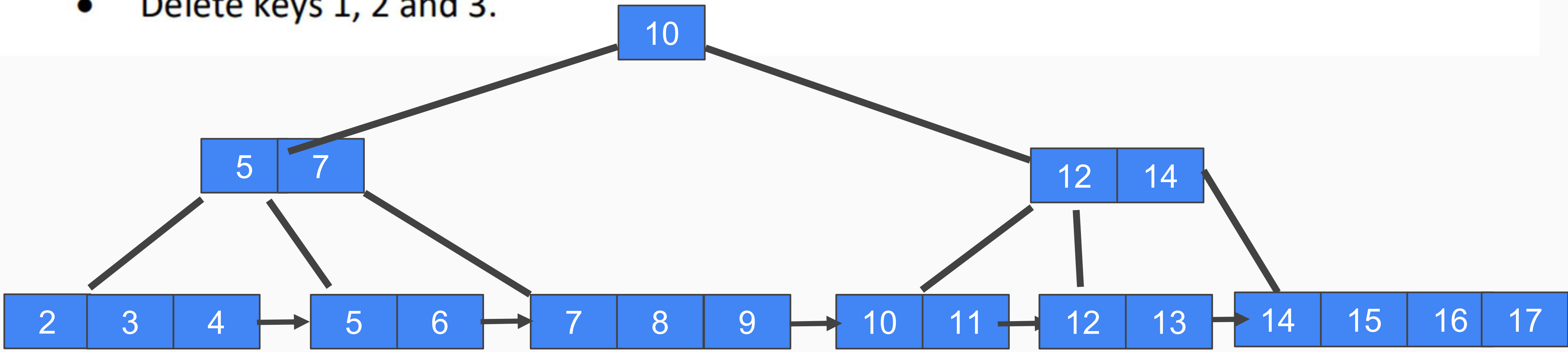
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Delete: 1

Question 2

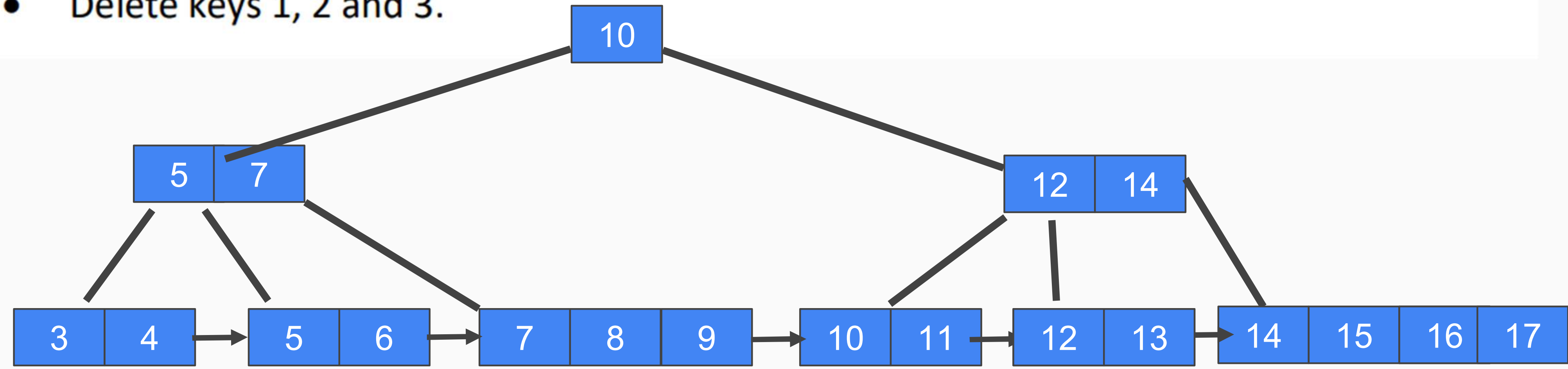
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Delete: 1

Question 2

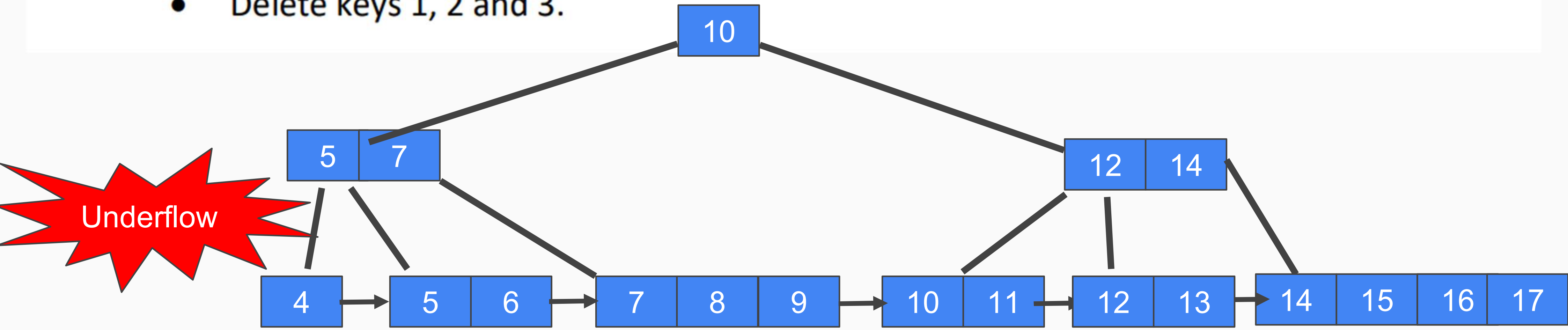
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Delete: 1, 2

Question 2

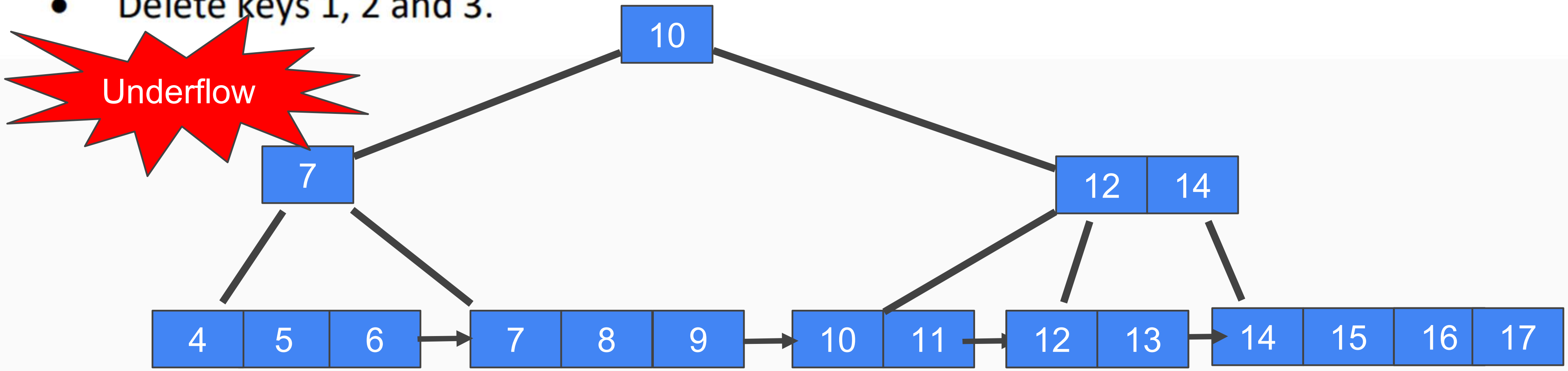
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
    - You insert the key 10 (and a pointer to this record) into the tree.
    - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
    - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
    - Delete keys 1, 2 and 3.



Delete: 1, 2, 3

Question 2

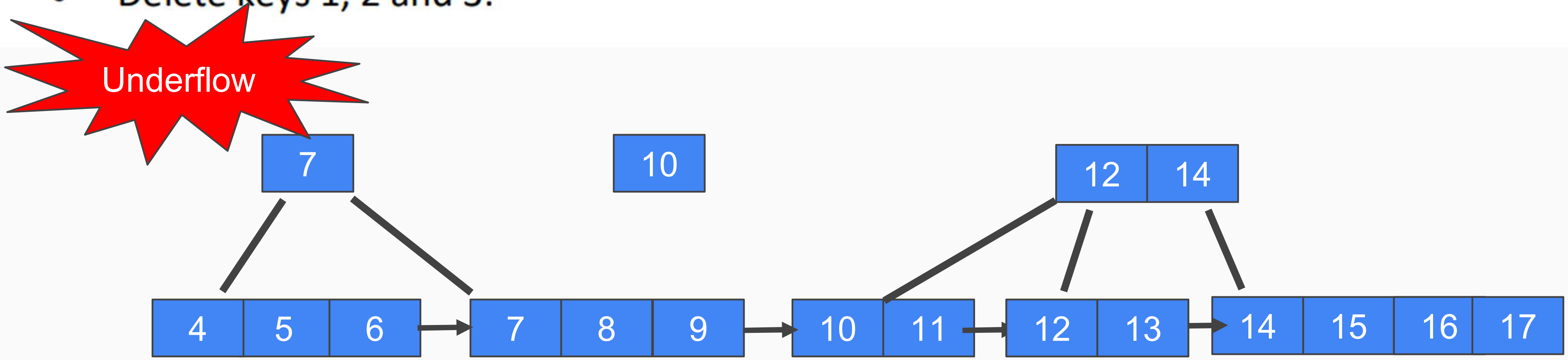
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Delete: 1, 2, 3

Question 2

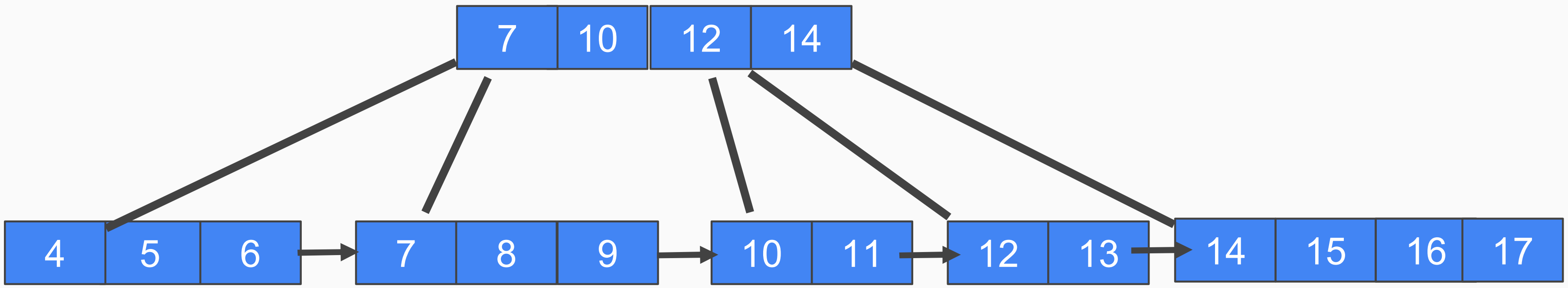
2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Delete: 1, 2, 3

Question 2

2. Consider a B+-tree of order 2 for this problem, that is initially empty. Show what the tree looks like at this point after each of the following actions:
   - You insert the key 10 (and a pointer to this record) into the tree.
   - Next you insert keys 2, 5, 14, and 17 (in this order) into the tree.
   - Next you insert keys 1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 15 and 16 (in this order).
   - Delete keys 1, 2 and 3.



Delete: 1, 2, 3

Question 2

Learning Objective:

1. Understand how the Insertion and Deletion algorithms of the B+ Tree Data Structure
2. Understand the conditions and the solutions to underflow / overflow of nodes

Question 3

3. To manage the buffer pool more efficiently, a database buffer manager might distinguish between different types of disk pages that may have different access patterns. In particular, it has been pointed out that index pages (e.g., B+-tree access) often have a different pattern of access compared to data pages. Consider the example index shown in the figure below:

```
A -->
    --> B -->
            --> C -->
                    --> D --> E
                          --> F
                          --> G
                          --> H
```

Node A is the root node of a B+-tree index, B, C, and D are index nodes, and E, F, G, H leaf nodes that contain the clustered data records. Consider the following logical page reference string corresponding to an index traversal

Question 3

$A_1, B_2, C_3, D_4, E_5, D_6, F_7, D_8, G_9, D_{10}, H_{11}, D_{12}, C_{13}, \ldots$

The subscript in each page number denotes the reference sequence number.
Assume that the allocated buffer has 5 pages. Contrast the use of an LRU replacement policy and an Optimal replacement policy (i.e., one which victimizes the page with the longest expected time until its next reference.)

Reference	LRU strategy Least → Most frequently used	Optimal Strategy
5	A B C D E	A B C D E
6	A B C E D	A B C D E
7	B C E D F/A	A B C D F/E
8		
9		
10		
11		
12		
13		

(A buffer entry I/J means that a page replacement has occured with page I replacing page J.)
Complete the above table. Why is LRU not an effective replacement algorithm here. Can you suggest a better strategy than LRU?

Question 3

$A_1, B_2, C_3, D_4, E_5, D_6, F_7, D_8, G_9, D_{10}, H_{11}, D_{12}, C_{13}, \ldots$

Ref	LRU	Changes	Optimal	Changes
5	A B C D E	0	A B C D E	0
6	A B C E D	0	A B C D E	0
7	B C D E **F/ A**	1	A B C D **F / E**	1
8	B C E F D	1	A B C D F	1
9	C E F D **G / B**	2	A B C D **G / F**	2
10	C E F G D	2	A B C D G	2
11	E F G D **H / C**	3	A B C D **H / G**	3
12	E F G H D	3	A B C D H	3
13	F G H D **C / E**	4	A N C D H	3

Question 3

$A_1, B_2, C_3, D_4, E_5, D_6, F_7, D_8, G_9, D_{10}, H_{11}, D_{12}, C_{13}, \ldots$

Ref	MRU	Changes	Optimal	Changes
5	A B C D E	0	A B C D E	0
6	A B C E D	0	A B C D E	0
7	A B C E **F / D**	1	A B C D **F / E**	1
8	A B C E **D / F**	2	A B C D F	1

Question 3

$A_1, B_2, C_3, D_4, E_5, D_6, F_7, D_8, G_9, D_{10}, H_{11}, D_{12}, C_{13}, \ldots$

Possible Solutions?

1. Replace based on the level of the B+ Tree by allocating buffer for leaf nodes and non-leaf nodes

Question 3

Learning Objective:

1. No Single Buffer replacement strategy will work best (incomplete information of future transactions)