Problem Set 2: Part I

Problem 1: Fixed-length and variable-length records

1.1 and 1.2

record contents

15172	Barbie#----------	2023	PG-13

length in bytes

34

show how you computed the length:

For fixed-length records, the stored values must take up the amount of space specified in type definition. So, the length in byte = 5 + 20 + 4 (4 bytes of int) + 5 = 34.

1.3 and 1.4

record contents

15172	#	Barbie	#	2023	#	PG-13	#

length in bytes

28

show how you computed the length:

For variable-length records, I choose to terminate field values with a special delimiter character '#', each of which is 2 bytes. Therefore, the length in byte = 5 + 2 + 6 + 2 + 4 + 2 + 5 + 2 = 28.

1.5 and 1.6

record contents

10	15	21	25	30	15172	Barbie	2023	PG-13

length in bytes

30

show how you computed the length:

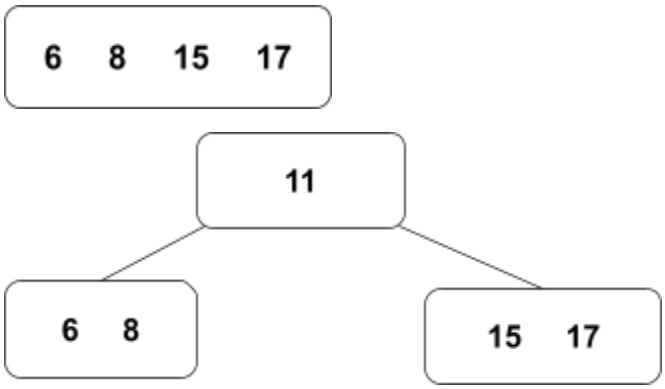
For variable-length records that begins with a header of offsets, there are (#fields+1) headers, i.e. 4+1 = 5 headers. Given that each integer metadata takes up 2 bytes, then the value of the first field will start at position 10, the second at position 10+len('15172')=15, etc. The last header offset stores the end position (the length) of this record, which is 30.

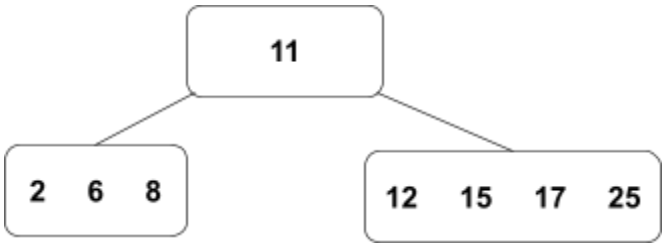
1.7

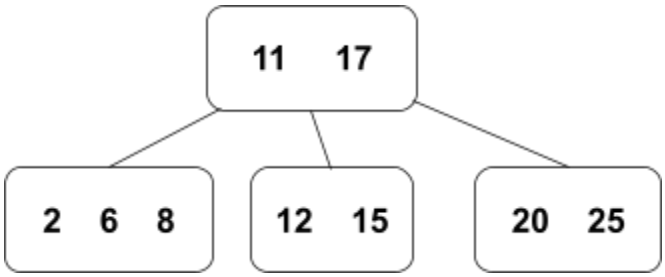
record contents

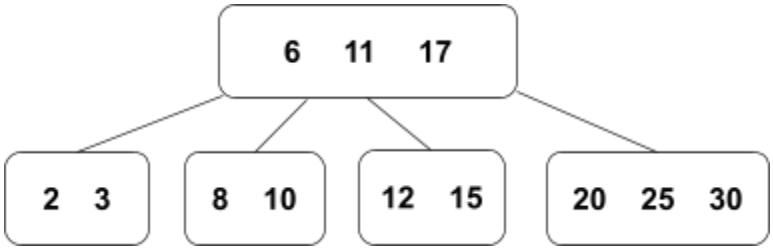
10	15	-1	31	36	87654	The Color Purple	PG-13

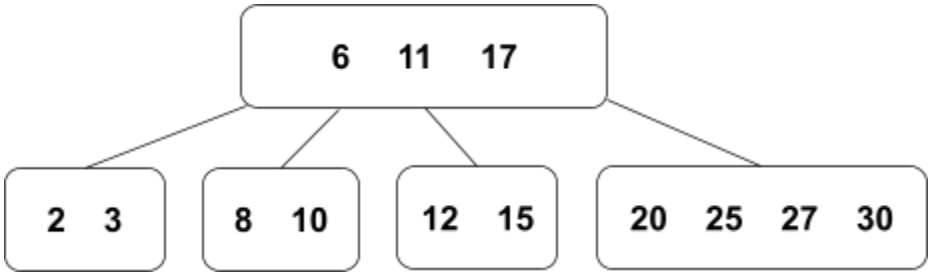
Problem 2.1: Insertions into a B-tree (order = 2)



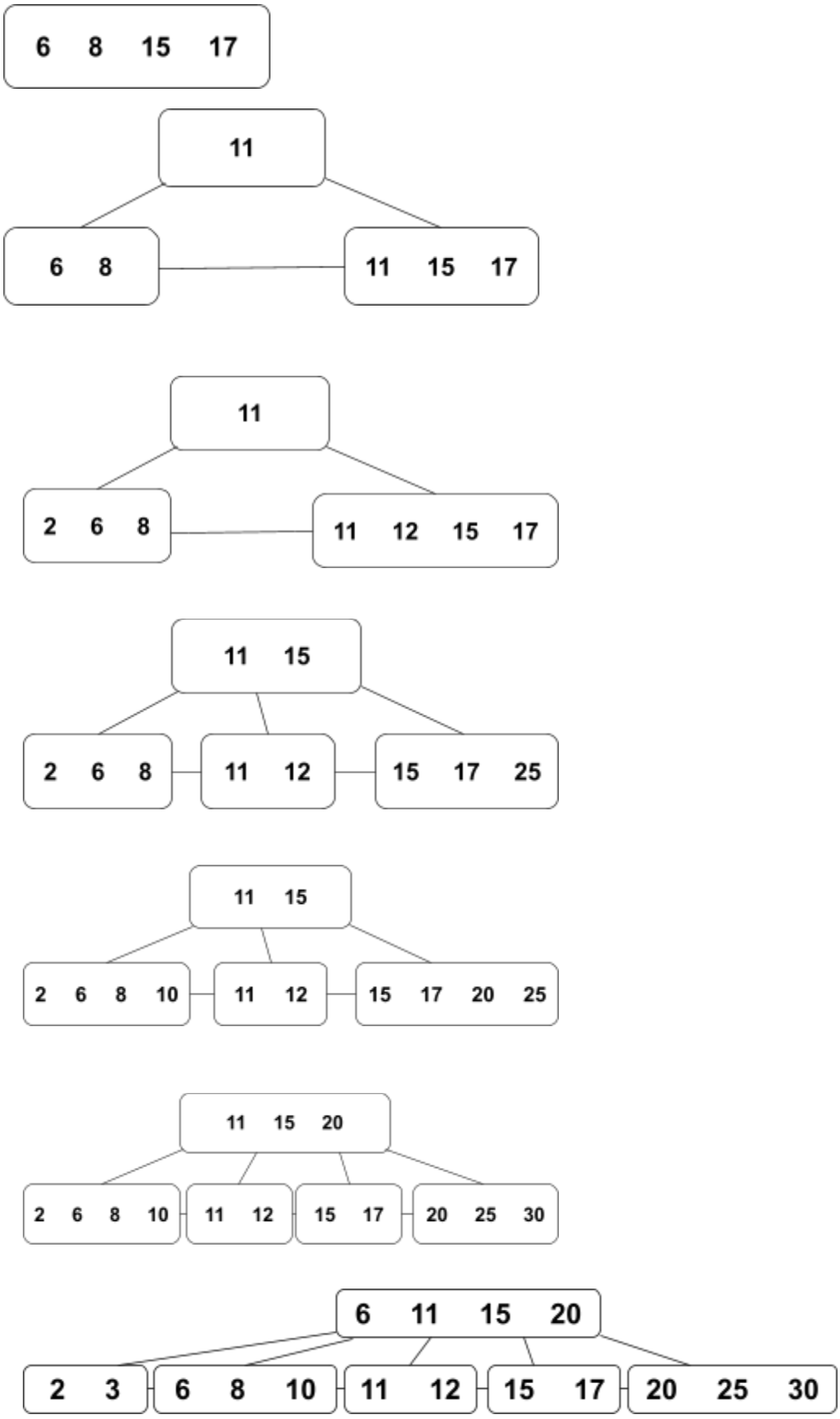


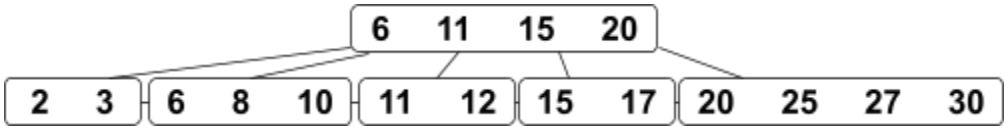






Problem 2.2: Insertions into a B+tree




6 11 15 20
2 3
6 8 10
11 12
15 17
20 25 27 30

Problem 2.3: Insertions into a linear hash table

a bucket is added whenever the number of items in the table exceeds three times the number of buckets.

6 = 0110
15 = 1111
8 = 1000
17 = 0001 0001
11 = 1011
12 = 1100
2 = 0010

25 = 0001 1001
20 = 0001 0100
10 = 1010

30 = 0001 1110
3 = 0011
27 = 0001 1011

i = ceil(log2n) = ceil(log2(5)) = 3 (11)

before first increase

00	6, 8, 12
01	15, 17, 11

after first increase

00	8, 12
01	15, 17, 11
10	6, 2

before second increase

00	8, 12, 20
01	15, 17, 11, 25
10	6, 2

after second increase

00	8, 12, 20
01	17, 25
10	6, 2, 10
11	15, 11

before third increase

00	8, 12, 20
01	17, 25
10	6, 2, 10, 30
11	15, 11, 3

after third increase

00	8
01	17, 25
10	6, 2, 10, 30
11	15, 11, 3, 27
100	12, 20

final state of the table

0	8
1	17, 25
2	6, 2, 10, 30
3	15, 11, 3, 27
4	12, 20