Problem Set 2: Part I

Problem 1: Fixed-length and variable-length records

1.1 and 1.2
record contents

748620	Paul Rudd#----------	04/06/1969	1

length in bytes

40

show how you computed the length:

len(id) + max_len(name) + len(dob) + len(num_kids)

6 + 20 + 10 + 4 = 40

1.3 and 1.4
record contents

6	748620	9	Paul Rudd	10	04/06/1969	4	1

length in bytes

37

show how you computed the length:

len(id) + len('Paul Rudd') + len(dob) + len(num_kids) + metadata

6 + 9 + 10 + 4 + 4*2 = 37

1.5 and 1.6
record contents

10	16	25	35	39	748620	Paul Rudd	04/06/1969	1

length in bytes

39

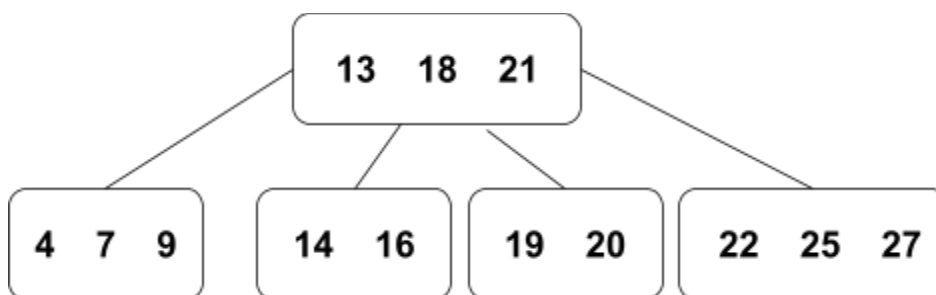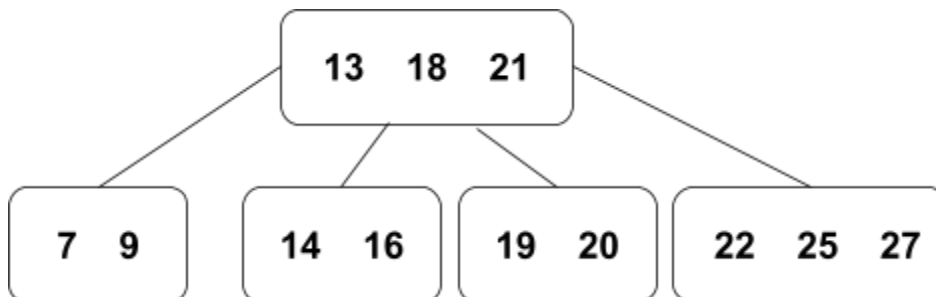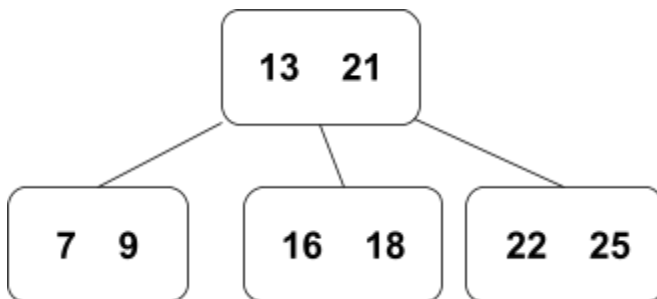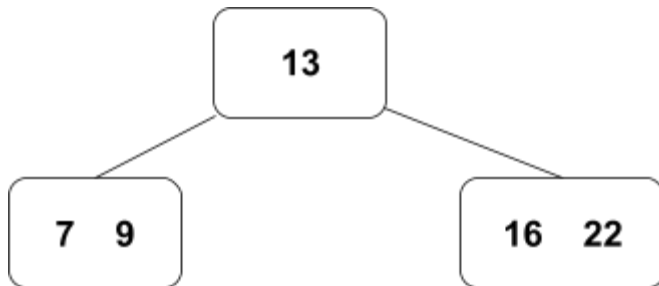show how you computed the length:

len(id) + len('Paul Rudd') + len(dob) + len(num_kids) + metadata

6 + 9 + 10 + 4 + 5*2 = 39

1.7

record contents

| 10 | 16 | -1 | 24 | 28 | 151726 | Issa Rae | 0 |

Problem 2.1: Insertions into a B-tree

Problem 2.2: Insertions into a B+tree

```
                    [ 13 ]
                   /      \
            [ 7  9 ]      [ 13  16  22 ]
```

```
                 [ 13  21 ]
                /    |     \
         [ 7  9 ] [ 13  16 ] [ 21  22  25 ]
```

```
                [ 13  16  21 ]
               /    |    |    \
        [ 7  9 ] [ 13  14 ] [ 16  18  20 ] [ 21  22  25 ]
```

```
                [ 13  16  21 ]
               /    |    |    \
   [ 4  7  9 ] [ 13  14 ] [ 16  18  19  20 ] [ 21  22  25  27 ]
```
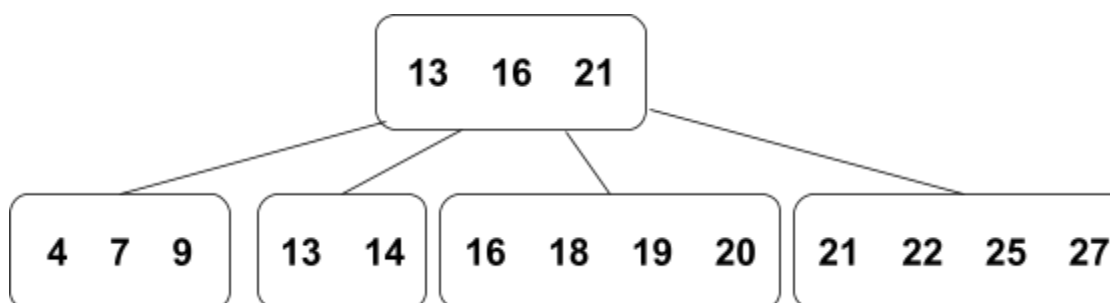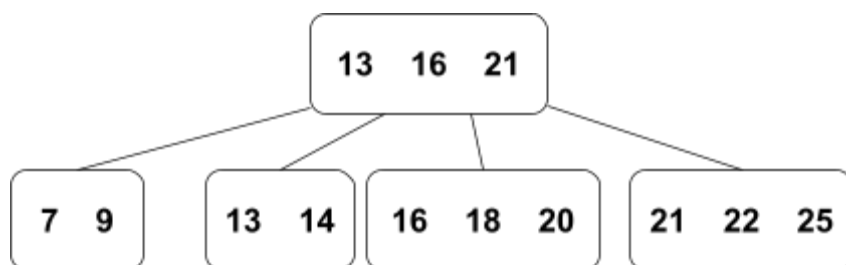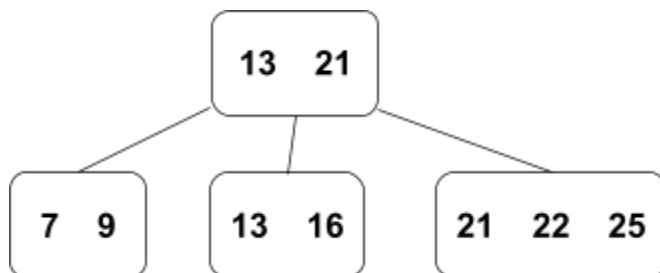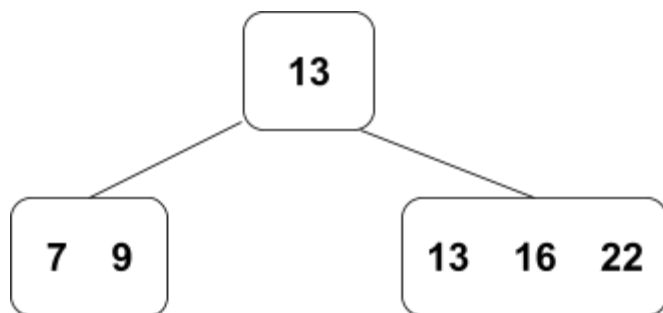
Problem 2.3: Insertions into a linear hash table

before first increase

0	16 22
1	9 13 7 21 25

after first increase

0	16
1	9 13 7 21 25
2	22

before second increase

0	16 20
1	9 13 7 21 25
2	22 18 14

after second increase

0	16 20
1	9 13 21 25
2	22 18 14
3	7

before third increase

0	16 20 4
1	9 13 21 25
2	22 18 14
3	7 27 19

after third increase

0	16
1	9 13 21 25
2	22 18 14
3	7 27 19
4	20 4

final state of the table

0	16
1	9 13 21 25
2	22 18 14
3	7 27 19
4	20 4