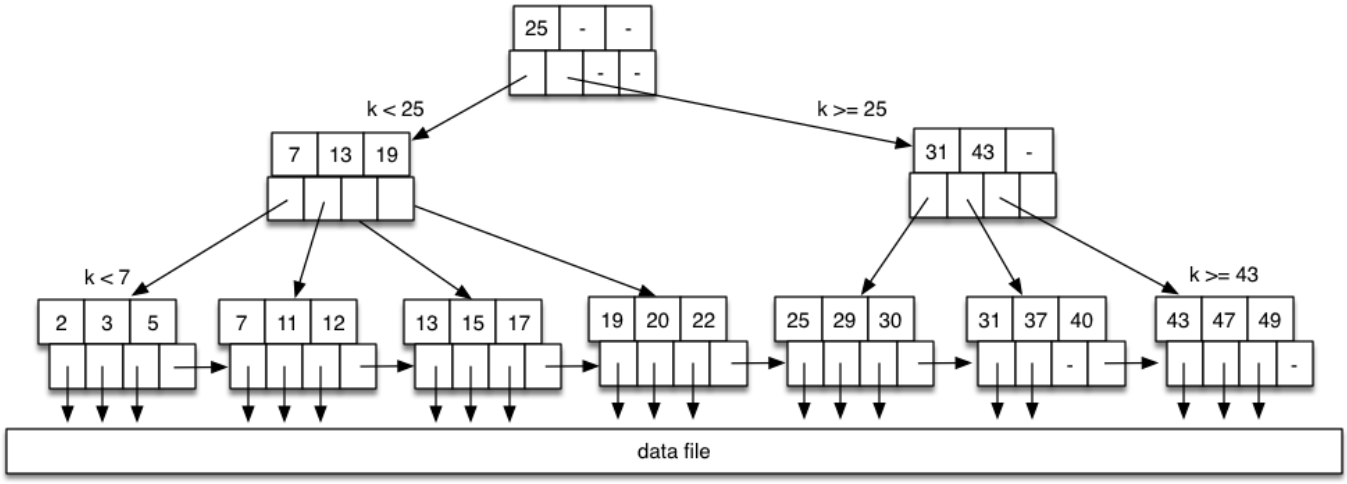
Quiz 3

Deadline	Friday, 02 April 2021 at 9:00PM
Latest Submission	Thursday, 01 April 2021 at 2:51AM
Raw Mark	4.00/4.00 (100.00%)
Late Penalty	N/A
Final Mark	4.00/4.00 (100.00%)

Question 1 (1 mark)

Consider the following B-tree



If the key value 26 is inserted into this tree, what will be the final value(s) in the root node?

If a node needs to be split, assume that the original middle value is the one promoted (e.g. if [2,3,5] was split, then 3 would be promoted).

(a) ◉	25 only
(b) ○	25 and 43
(c) ○	25 and 31
(d) ○	25 and 26
(e) ○	None of the other options is correct.

✔ Your response was correct.
Mark: 1.00

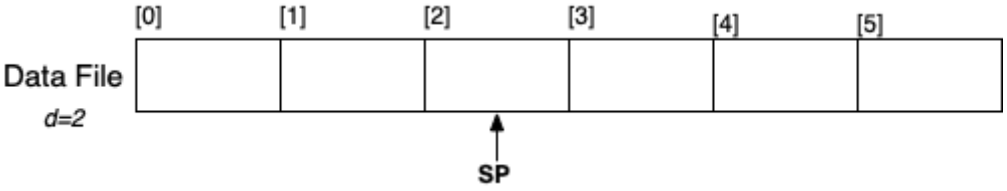
Traversal of the tree takes us to the (25,29,30) node. This node is full, so we need to promote 29 to the parent node and split into two new leave (25,26) and (29,30).

There is space for the promoted key value 29 in the (31,43) node, so 29 is added there and we adjust the links to the child nodes.

No further promotion occurs, so the root node remains unchanged at (25,-,-)

Question 2 (1 mark)

Consider the following linear-hashed file with depth *d=2* and the split pointer at page 2. Assume that it has no overflow pages.



If a query with hash value `...01110101` is run on this table, which page(s) will need to be read to answer the query?

(a) ○	page 1 only
(b) ○	page 2 only
(c) ◉	page 5 only
(d) ○	pages 1 and 2
(e) ○	pages 1 and 5
(f) ○	None of the other options is correct

✔ Your response was correct.
Mark: 1.00

Since the last 2 bits of the hash value are `....01` , and this gives a page address which is less than *sp* , we need to consider 3 bits of the hash value, giving a hash of `...101` , which is page 5. So we read page 5 only.

Question 3 (1 mark)

Consider a relation *R(a,b,c)* implemented as a multi-attribute hashed file with the following parameters:

b =1024 data pages
d =10 bits for hash values
d_a=5 bits contributed by *a*
d_b=3 bits contributed by *b*
d_c=2 bits contributed by *c*

Assuming that there are no overflow pages, how many pages will be fetched in answering the query:

```
select * from R where a=3 and c=1;
```

(a) ○	3
(b) ◉	8
(c) ○	32
(d) ○	1024
(e) ○	None of the other options is correct.

✔ Your response was correct.
Mark: 1.00

If we have known values for attributes **a** and **c** , that gives us 7 known hash bits.

There are 3 unknown hash bits from attribute **b** . We need to consider all possible combinations of 0's and 1's for these 3 bits, so 8 possibilities.

Question 4 (1 mark)

Consider a table defined as:

```
create table Employees (
   id integer primary key,
   name varchar(50),
   address varchar(100), ...
);
create index on table Employees (id);
```

Employee records are added to the table as employees are hired, and never removed. Thus, the file holding the data pages of the `Employees` table has data pages packed as full as possible, with no overflow pages, and the file is sorted on the `id` attribute. The table has the following characteristics:

B =8192 bytes per page (both for data pages and index pages)
r =10000 total tuples
R =200 bytes (average, but assume all tuples are this length)
R_i=8 bytes (each index tuple contains (Employee.id,pageID)

Assuming that data pages have a 192-byte header and and index pages have a 32-byte header, and that a sparse index is used, what are the values of *b* (the total number of data pages) and *i* (the total number of index pages)? The choices below are presented as *(b,i)* pairs.

(a) ◉	(250, 1)
(b) ○	(250, 10)
(c) ○	(10000, 1)
(d) ○	(10000, 10)

(e) ○	None of the other options is correct.

✔ Your response was correct.
Mark: 1.00

The number of tuples that can be stored in each page is c = (8192-192)/200 = 40

The number of data pages b = ceil(r/c) = ceil(1000/40) = 250

The number of index entries in each index page is c_i = (8192-32)/8 = 1020.

Since we have one index entry for each data page (sparse index), we only need one page to hold all 250 index entries.