

1. In variable-length record representation, the record starts with offset and length pairs of variable-size attributes, followed by fixed-size attributes, then the null bitmap, and finally the variable-size attributes. How can we improve this representation if our application is expected to store tables with large number of attributes, most of which are nulls?
- store the null bitmap at the beginning then store the fixed-sized attributes then the two variable-size attributes. this should be efficient in space.

2. Consider the following arrangement for four disks, where Bi is a data block, and Pi is the parity block for the 4 data blocks that precedes it. What problem will this arrangement cause?
- A lost of data will occur if by any chance a disk gets damaged since the parity block is not found on B1 to B4 of disk 1 through 4.

3. Consider the following file organization using free list.
- | header | | | | |
|-----------|-------|------------|------------|-------|
| record 0 | 10101 | Srinivasan | Comp. Sci. | 65000 |
| record 1 | | | | |
| record 2 | 15151 | Mozart | Music | 40000 |
| record 3 | 22222 | Einstein | Physics | 95000 |
| record 4 | | | | |
| record 5 | 33456 | Gold | Physics | 87000 |
| record 6 | | | | |
| record 7 | 58583 | Califieri | History | 62000 |
| record 8 | 76543 | Singh | Finance | 80000 |
| record 9 | 76766 | Crick | Biology | 72000 |
| record 10 | 83821 | Brandt | Comp. Sci. | 92000 |
| record 11 | 98345 | Kim | Elec. Eng. | 80000 |
- Show the structure of the file after each of the following operations (they follow each other):
- (a) Delete record 9.

After deleting record 9, we will modify record 6 (the previously deleted record on the list) to point to the address of record 9 marking it as empty and to make record 9 point to a null.

header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9				
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

(b) Insert (20000, Jamie, Physics, 100000).

Find where we can add the entry, if there is a free space put the entry there. if not then we must insert it in another overflow block and update the pointer chain

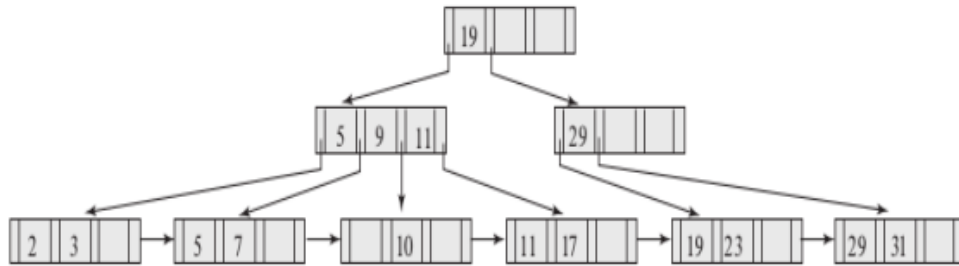
header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9				
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

20000	Jamie	Physics	100000
-------	-------	---------	--------

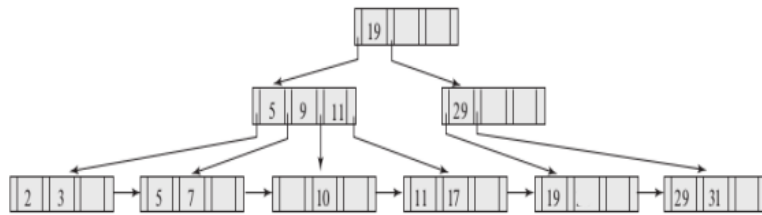
(I'm sorry)

4.		Construct a B+-tree for the following set of key values: (2, 3, 5, 7, 11, 17, 19, 23, 29, 31). The tree is initially empty and values are added one value at a time in ascending order. Consider the following values of n:
	(a)	<p>n = 4</p>
	(b)	<p>n = 6</p>
	(c)	<p>n = 8</p>

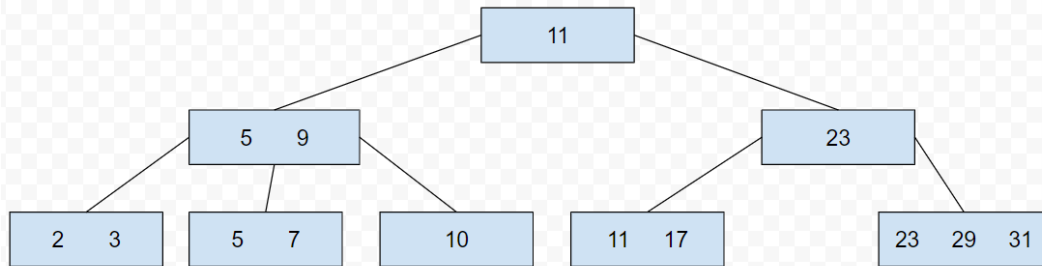
3.

Consider the following B^+ -tree with $n = 4$:

(a) Delete 23



(b) Delete 19



4.

Consider the following B^+ -tree with $n = 6$: Insert 8 into this tree.