1. Suppose Hash tables are built with the numbers inserted in the order of 12, 10, 26, 8, 2, 5, 40, 11.

Draw the hash tables by respectively adopting the following two combinations of hash function and collision resolution strategies.

* 1. Bucket hashing with a 10-slot hash table using the hash function ***h(K) = (3K+1) mod 10*.** Each bucket contains 2 slots. If a bucket is full, the record is stored in the overflow bucket.
  2. Closed hashing with a 10-slot hash table using ***h(K) = (3K+1) mod 10***; linear probing with the probe function *p(K, i) = 3i*

Answer:

(1)

Hash table overflow

|  |
| --- |
| 40 |
| 10 |
|  |
|  |
| 11 |
| 8 |
| 2 |
| 12 |
|  |
| 26 |

|  |
| --- |
| 5 |
|  |
|  |
|  |

(2)

|  |
| --- |
| 2 |
| 10 |
|  |
| 11 |
| 40 |
| 8 |
| 5 |
| 12 |
|  |
| 26 |

1. Given the following B+-tree of order 3, in which the leaf nodes can store up to 3 records.

Two records with key values 13 and 50 are inserted in the given order. Draw the updated B+-tree after each insertion, respectively.

10 12 14

20 31

42 43 54

20

42

Answer:

After inserting 13

13

42

10 12

20 31

42 43 49

13 14

20

After inserting 50

20

13

42

50

10 12

20 31

42 43

13 14

49 50

1. Suppose Prim’s algorithm is revised to find the maximum spanning tree (MaxST). The revision is simply that the MaxST is repeatedly expanded by selecting the largest-cost edge from a vertex currently in the MaxST to a vertex not currently in the MaxST.

Run the revised Prim’s algorithm described above on the following graph staring at vertex 2. Draw the final MaxST.



Answer：



1. Assume that a computer system has disk blocks of 4096 bytes, and a record have 4-byte keys and 12-byte data fields. There are 230 records which are sorted and packed sequentially into the disk file.
   1. Suppose a linear index is constructed to index the records, which uses 4 bytes to store the key and 4 bytes to store the block ID for the associated records. What is the size of the linear index?
   2. Suppose a B+-tree of order 128 is built with each leaf node corresponding to each data block. An internal node has up to 128 key/pointer pairs where the key size and the pointer size are both 4 bytes. Estimate the size of the smallest possible B+-tree with the size of the data blocks excluded.

Answer:

1. 1block stores 28=256 records; 230 records needs 230/28 = 222 blocks

For linear index, the size is 222\*(4+4) =225 bytes = 32MB

1. For the B+-tree index,

the number of internal nodes at the level above leaf nodes is 222/27=215

the number of internal nodes one level above is 215/27=28

the number of internal nodes one level above is 2

one root node

the size of an internal node is 128\*8=210 bytes = 1KB. In total, the size of B+-tree is （215+28+2+1）\*1KB=32.25MB