Exercise 1: Construct bulk loading B+ tree with order P = 3 and following sequence of keys: 2, 10, 15, 1, 3, 8, 12, 17, 35, 40, 60, 20, 21.

Answer. In this question, we follow the procedure of bottom up B+ tree construction. Sort the keys and form all nodes in the leaf level of the tree. The minimum value in each block, along with the pointer to the block, is used to create entries in the next level of the B+-tree, pointing to the leaf blocks. Each further level of the tree is similarly constructed using the minimum values associated with each node one level below, until the root is created. Please refer to page 504 of Korth 6th edition.

Root - > 20, 60 keys
Level 1 -> (3,10) , (20,35), (60) i.e. 3 nodes at level 1
Level 2 (leaf nodes) -> (1,2), (3,8), (10,12), (15,17), (20,21), (35,40), (60) i.e. 7 leaf nodes

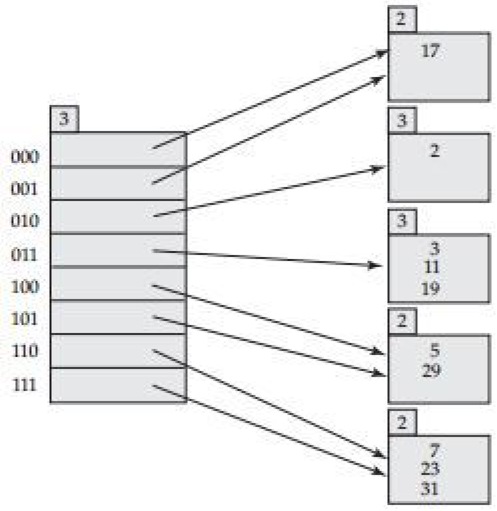
Exercise 2: Suppose that we are using extendable hashing on a file that contains records with the following search-key values:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31

Show the extendable hash structure for this file if the hash function is h(x) = x mod 8 and buckets can hold three records.

Answer.

Use the first *i* high order bits of *X* as a displacement into bucket address table, and follow the pointer to appropriate bucket. Please refer to the lecture slide 11.64.



Exercise 3: Show how the extendable hash structure of Exercise 3 changes as the result of each of the following steps:

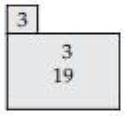
a. Delete 11.

b. Delete 31.

c. Insert 1.

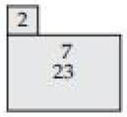
d. Insert 15.

Answer. Delete 11: From the answer to Exercise 2, change the third bucket to:

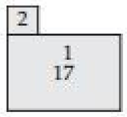


At this stage, it is possible to coalesce the second and third buckets. Then it is enough if the bucket address table has just four entries instead of eight. For the purpose of this answer, we do not do the coalescing.

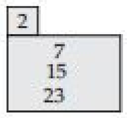
Delete 31: From the answer to Exercise 2, change the last bucket to:



Insert 1: From the answer to Exercise 3, change the first bucket to:

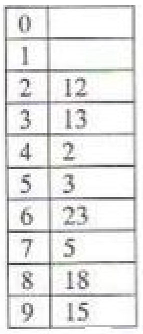


Insert 15: From the answer to Exercise 3, change the first bucket to:



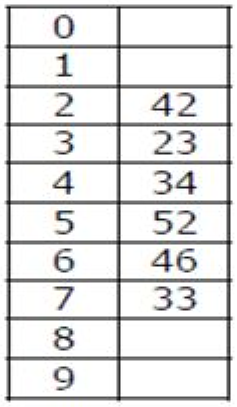
Exercise 4: The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?

Answer.



0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

Exercise 5: A hash table of length 10 uses open addressing with hash function h(k)=k mod 10 and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.



0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Write the possible order in which the key values could have been inserted in the table?

Answer.

46, 34, 42, 23, 52, 33

Exercise 6: Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4) mod 7. Assuming the hash table is initially empty, what will be the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing?

Answer.

0	1	2	3	4	5	6
1	8	10				3

Exercise 7: Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length m = 11 using open addressing with the auxiliary hash function h'(k) = k. Illustrate the result of inserting these keys using quadratic probing with $c_1 = 1$ and $c_2 = 3$.

Answer. With quadratic hashing, we use the hash function
$h(k, i) = (h'(k) + i + 3i^2) \bmod m = (k + i + 3i^2) \bmod m$.
h(10, 0) = (10 + 0 + 0) mod 11 = 10. Thus we have T[10] = 10.
h(22, 0) = (22 + 0 + 0) mod 11 = 0. Thus we have T[0] = 22.
h(31, 0) = (31 + 0 + 0) mod 11 = 9. Thus T[9] = 31.
h(4, 0) = (4 + 0 + 0) mod 11 = 4. Thus T[4] = 4.
h(15, 0) = (15 + 0 + 0) mod 11 = 4. Since T[4] is occupied, we probe again, h(15,1) = (15 + 1 + 3) mod 11 = 8. Thus T[8] = 15.
h(28, 0) = (28 + 0 + 0) mod 11 = 6. Thus T[6] = 28.
h(17, 0) = (17 + 0 + 0) mod 11 = 6. Since T[6] is occupied, we probe again. h(17, 1) = 10. We probe again as T[10] is occupied. Similarly h(17,2) = 9 does not work. Finally the insert succeeds with h(17,3) = 3. Thus T[3] = 17.
h(88, 0) = (88 + 0 + 0) mod 11 = 0. T[0] is occupied, so probe again. h(88,1) = 4, h(88,2)=3, h(88,3) = 8, h(88,4) = 8, h(88,5) = 3, h(88,6) = 4, h(88,7) = 0 do not work. We finally succeed with h(88,8) = 2. Thus T[2] = 88.
h(59,0) = (59 + 0 + 0) mod 11 = 4. T[4] is occupied. h(59,1) = (59 + 1 + 3) mod 11 = 63 mod 11 = 8. T[8] occupied. h(59,2) = (59 + 2 + 12) mod 11 = 7. T[7] = 59.

0	1	2	3	4	5	6	7	8	9	10
22		88	17	4		28	59	15	31	10

Exercise 8: Consider the following schema.

Enroll(StudentID, CourseID, Term, Grade)

Suppose we want to find the maximum grade of the CSE202 class in Winter 2002 term. Assume that there is no other index on Enroll. Which one of the following four options will likely provide the biggest performance improvement for the query? Briefly justify your answer for each option.

(A) Create a clustered index on Enroll(StudentID)
(B) Create an index on Enroll(StudentID) and another index on Enroll(Grade) and none of the index is clustered.
(C) Create a clustered index on Enroll(CourseID)and another index on Enroll(Term)
(D) Create a secondary index on Enroll(CourseID, Term)

Answer. Best Option: C) Create a cluster index on Enroll(courseId) and another index on Enroll(Term)
By using the index on courseID attribute get the block addresses for CSE 202 course. Use the index on attribute Term to get the block addresses that have Winter2002 term records. Then intersection of these block addresses would give us the desired blocks addresses. Start scanning the file from the block address with the least course ID in the increasing order of block addresses to get the result records. The number of Disk IOs will be less than equal to the number of result tuples in addition to index accesses. However, generally it is going to be very less as the index is clustered on courseID and hence most of the result records will be highly likely together on the same blocks.

Explanation:
Option (A) is not good as it will result into linear scan of complete file.
Option(B) is not good as indices do not help at all. If any or both indices are used, the number of Disk IOs will be equal to number of records in addition to accesses made for index structure accesses.
Option(D) is not good as the number of Disk IOs will be equal to the number of students enrolled for CSE202 course in Winter 2002 term, i.e. the number of tuples which satisfy this condition.