R-20.3  For what values of d is the tree T of the previous exercise an order-d B-tree?

**Solution:** From previous exercise T is a valid (a, b) tree for (4, 8) or (5, 9) tree.

The value order d B-tree in (a, b) with a= d/2 and b=d

Taking (4, 8), the value of d will be 8.

Taking (5, 9), the value of d will be 9.

A-20.1  Suppose you are processing a large number of operations in a consumer-producer process, such as a buffer for a large media stream. Describe an external-memory data structure to implement a queue so that the total number of disk transfers needed to process a sequence of n enqueue and dequeue operations is O(n/B).

**Solution:** We can use linked list to implement a queue so that total number of disk transfer needed to process a sequence of n enqueue and dequeue operation.

As inserting (enqueue) an element at the end of the list will use O(1) disk transfers.

for dequeuing elements from the front, it will return the block to the free memory heap when it becomes empty.

This will perform O(n/B) disk transfers.

A-20.2  Imagine that you are trying to construct a minimum spanning tree for a large net work, such as is defined by a popular social networking website. Based on using Kruskal’s algorithm, the bottleneck is the maintenance of a union-find data structure. Describe how to use a B-tree to implement a union-find data structure (from Section 7.1) so that union and find operations each use at most O(log n/ log B) disk transfers each.

**Solution:**

Implementing union-find data structure using B-tree. Suppose, initially all the nodes are in singleton trees(having height 1) The height of the tree increases by 1 when a node attached with the larger group and the number of nodes in the tree is doubled atleast.

Maximum number of nodes in any tree is n, so the height of the resulting tree can be at most logn.

So find operation will take O(log n) because visiting O(log n) nodes and each union operation will take O(1) and performing O(log n)union will take O(log n)

Implementing union-find data structure using B-tree with n items executes O()disk transfers in union and find operation.

R-23.11 What is the longest prefix of the string "cgtacgttcgtacg" that is also a suffix of this string?

**Solution:**

The longest prefix of the string that is also a suffix of the string is “cgtacg”

C-23.1 Give an example of a text T of length n and a pattern P of length m that force the brute-force pattern matching algorithm to have a running time that is Ω(nm).

**Solution**: Consider a test T of length *n* as AAAAAAAAA………….AS

Now consider a pattern P of length *m* as AAAS

While comparing each letter of pattern P in test string T, the match will found at the end of the string. The worst case running time for brute force is *mn* then the running time will be *Ω(nm).*

A-23.5  One way to mask a message, M, using a version of *steganography*, is to insert random characters into M at pseudo-random locations so as to expand M into a larger string, C. For instance, the message, could be expanded into

ILOVEMOM,

AMIJLONDPVGEMRPIOM.

It is an example of hiding the string, M, in plain sight, since the characters in M and C are not encrypted. As long as someone knows where the random characters where inserted, he or she can recover M from C. The challenge for law enforcement, therefore, is to prove when someone is using this technique, that is, to determine whether a string C contains a message M in this way. Thus, describe an O(n)-time method for detecting if a string, M, is a subsequence of a string, C, of length n.

**Solution:**

We traverse both the strings M and C say left to right. If we find a matching character, we will increment the pointer in both the strings. Otherwise we will increment the pointer of string C only. If we traverse all the characters of M that means, M is a subsequence of C

Time Complexity: The time complexity will be O(n) as we traverse the string C which is of length n.