1BH18CS053
Neelesh [illegible]

18/11/24

B tree Insertions

Algorithm

```
BTree Insertion (T,K)

r root[T]
  if n[r] = 2t - 1
       s= Allocate Node()
         root[T] = s
         leaf[S] = FALSE
      n[S] ← 0
      c1[S] ← r
       Btree SplitChild (S, 1, r)
       BtreeInsertNonFull (S, K)
  else
       Btree Insert NonFull (r,k)
Btree InsertNonFull(x, k)
    i = n(x)
    if leaf (x)
        while i ≥ 1 and k < keyi[x]
              keyi+1[x] = keyi[x]
                  i = i - 1
              keyi+1[x) = k
                  n[x] = n[x] + 1
        else while i ≥ 1 and k < keyi(x)
                  i = i - 1
             i = i + 1
        if n [ci[x]] = = 2t - 1
         Btree SplitChild (x, i, ci[x])
           if k > keyi[x]
              i = i+1
```

```
B-tree Insert Nonfull      (ci(x), k)
    Btree Split Child (x, i)
    Btree Sput Chid (x, i, y)
        Z = Allocate Node()
    leaf [z] = leaf (y)
        n[z] = t-1
        for j = 1 to t-1
            key j [z] = key j + t(y)
      if not leaf [y]
          for j = 1 to t
              c j [z] = c j + t(y)
    n[y] = t-1
        for j = n(x)+1 to i+1
          c j+1 (x) = c j (x)
          c i+1 [x] = z
          for j = n(x) to i
              key (j+1) (x) = key j (x)
      key i (x) = key t (y)
          n[x] = n(x)+1
```