화일처리 보고서
B-Tree 삽입

20181653 이강희

November 15, 2019

강의 슬라이드를 참조하였습니다.

1 출력 결과

output.txt

```
  File Processing
      B-Tree
=================
M = 3
Insert 40        40
Insert 11        11 40
Insert 77        11 40 77
Insert 33        11 33 40 77
Insert 20        11 20 33 40 77
Insert 90        11 20 33 40 77 90
Insert 99        11 20 33 40 77 90 99
Insert 70        11 20 33 40 70 77 90 99
Insert 88        11 20 33 40 70 77 88 90 99
Insert 80        11 20 33 40 70 77 80 88 90 99
Insert 66        11 20 33 40 66 70 77 80 88 90 99
Insert 10        10 11 20 33 40 66 70 77 80 88 90 99
Insert 22        10 11 20 22 33 40 66 70 77 80 88 90 99
Insert 30        10 11 20 22 30 33 40 66 70 77 80 88 90 99
Insert 44        10 11 20 22 30 33 40 44 66 70 77 80 88 90 99
Insert 55        10 11 20 22 30 33 40 44 55 66 70 77 80 88 90 99
Insert 50        10 11 20 22 30 33 40 44 50 55 66 70 77 80 88 90 99
Insert 60        10 11 20 22 30 33 40 44 50 55 60 66 70 77 80 88 90 99
Insert 100       10 11 20 22 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 28        10 11 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 18        10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 9         9 10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 5         5 9 10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 17        5 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 6         5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 3         3 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 1         1 3 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 4         1 3 4 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 2         1 2 3 4 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 7         1 2 3 4 5 6 7 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 8         1 2 3 4 5 6 7 8 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 73        1 2 3 4 5 6 7 8 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 12        1 2 3 4 5 6 7 8 9 10 11 12 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 13        1 2 3 4 5 6 7 8 9 10 11 12 13 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 14        1 2 3 4 5 6 7 8 9 10 11 12 13 14 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 16        1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 15        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 25        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 24        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 28        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 45        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 45 50 55 60 66 70 73 77 80 88 90 99 100
Insert 49        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 45 49 50 55 60 66 70 73 77 80 88 90 99
                 100
Insert 42        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 42 44 45 49 50 55 60 66 70 73 77 80 88 90 99
                 100
Insert 43        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 42 43 44 45 49 50 55 60 66 70 73 77 80 88 90
                 99 100
Insert 41        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 49 50 55 60 66 70 73 77 80 88
                 90 99 100
Insert 47        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 47 49 50 55 60 66 70 73 77 80
                 88 90 99 100
Insert 48        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 47 48 49 50 55 60 66 70 73 77
                 80 88 90 99 100
Insert 46        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 66 70 73
                 77 80 88 90 99 100
Insert 63        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 63 66 70
                 73 77 80 88 90 99 100
Insert 68        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 63 66 68
                 70 73 77 80 88 90 99 100
Insert 61        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 63 66
                 68 70 73 77 80 88 90 99 100
Insert 62        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                 66 68 70 73 77 80 88 90 99 100
Insert 64        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                 64 66 68 70 73 77 80 88 90 99 100
Insert 69        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                 64 66 68 69 70 73 77 80 88 90 99 100
Insert 67        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                 64 66 67 68 69 70 73 77 80 88 90 99 100
Insert 65        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 54        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 60 61 62
                 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 59        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 59 60 61
                 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 58        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 58 59 60
                 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 51        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 54 55 58 59
                 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 53        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 53 54 55 58
                 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 57        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 53 54 55 57
                 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 52        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 56        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 83        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 83 88 90 99 100
Insert 81        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 83 88 90 99 100
Insert 82        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 82 83 88 90 99 100
Insert 84        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 82 83 84 88 90 99 100
Insert 75        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 75 77 80 81 82 83 84 88 90 99 100
Insert 89        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 75 77 80 81 82 83 84 88 89 90 99 100
=================
M = 4
Insert 40        40
```

```
Insert 11       11 40
Insert 77       11 40 77
Insert 33       11 33 40 77
Insert 20       11 20 33 40 77
Insert 90       11 20 33 40 77 90
Insert 99       11 20 33 40 77 90 99
Insert 70       11 20 33 40 70 77 90 99
Insert 88       11 20 33 40 70 77 88 90 99
Insert 80       11 20 33 40 70 77 80 88 90 99
Insert 66       11 20 33 40 66 70 77 80 88 90 99
Insert 10       10 11 20 33 40 66 70 77 80 88 90 99
Insert 22       10 11 20 22 33 40 66 70 77 80 88 90 99
Insert 30       10 11 20 22 30 33 40 66 70 77 80 88 90 99
Insert 44       10 11 20 22 30 33 40 44 66 70 77 80 88 90 99
Insert 55       10 11 20 22 30 33 40 44 55 66 70 77 80 88 90 99
Insert 50       10 11 20 22 30 33 40 44 50 55 66 70 77 80 88 90 99
Insert 60       10 11 20 22 30 33 40 44 50 55 60 66 70 77 80 88 90 99
Insert 100      10 11 20 22 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 28       10 11 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 18       10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 9        9 10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 5        5 9 10 11 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 17       5 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 6        5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 3        3 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 1        1 3 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 4        1 3 4 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 2        1 2 3 4 5 6 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 7        1 2 3 4 5 6 7 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 8        1 2 3 4 5 6 7 8 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 77 80 88 90 99 100
Insert 73       1 2 3 4 5 6 7 8 9 10 11 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 12       1 2 3 4 5 6 7 8 9 10 11 12 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 13       1 2 3 4 5 6 7 8 9 10 11 12 13 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 14       1 2 3 4 5 6 7 8 9 10 11 12 13 14 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 16       1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 15       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 25       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 24       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 28       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 50 55 60 66 70 73 77 80 88 90 99 100
Insert 45       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 45 50 55 60 66 70 73 77 80 88 90 99 100
Insert 49       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 44 45 49 50 55 60 66 70 73 77 80 88 90 99
                100
Insert 42       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 42 44 45 49 50 55 60 66 70 73 77 80 88 90 99
                100
Insert 43       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 42 43 44 45 49 50 55 60 66 70 73 77 80 88 90
                99 100
Insert 41       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 49 50 55 60 66 70 73 77 80 88
                90 99 100
Insert 47       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 47 49 50 55 60 66 70 73 77 80
                88 90 99 100
Insert 48       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 47 48 49 50 55 60 66 70 73 77
                80 88 90 99 100
Insert 46       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 66 70 73
                77 80 88 90 99 100
Insert 63       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 63 66 70
                73 77 80 88 90 99 100
Insert 68       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 63 66 68
                70 73 77 80 88 90 99 100
Insert 61       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 63 66
                68 70 73 77 80 88 90 99 100
Insert 62       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                66 68 70 73 77 80 88 90 99 100
Insert 64       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                64 66 68 70 73 77 80 88 90 99 100
Insert 69       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                64 66 68 69 70 73 77 80 88 90 99 100
Insert 67       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                64 66 67 68 69 70 73 77 80 88 90 99 100
Insert 65       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 55 60 61 62 63
                64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 54       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 60 61 62
                63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 59       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 59 60 61
                62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 58       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 54 55 58 59 60
                61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 51       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 54 55 58 59
                60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 53       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 53 54 55 58
                59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 57       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 53 54 55 57
                58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 52       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 56       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 88 90 99 100
Insert 83       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 83 88 90 99 100
Insert 81       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 83 88 90 99 100
Insert 82       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 82 83 88 90 99 100
Insert 84       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 77 80 81 82 83 84 88 90 99 100
Insert 75       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 75 77 80 81 82 83 84 88 90 99 100
Insert 89       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 25 28 30 33 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
                56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 73 75 77 80 81 82 83 84 88 89 90 99 100
```

2 소스 코드

```
1 #include <iostream>
2 #include <vector>
```

```cpp
#include <stack>
#include <algorithm>
using namespace std;

class BTreeNode {
public:
    BTreeNode(int key) {
        keys.push_back(key);
        childs.assign(2, nullptr);
        n = 1;
    }
    BTreeNode(const BTreeNode *node) : n(node->n), keys(node->keys), childs(node->childs) {}
    ~BTreeNode() {
        for (auto child : childs) delete child;
    }
    void insertKey(int newKey, BTreeNode *newChild=nullptr) {
        keys.push_back(newKey);
        childs.push_back(newChild);
        int i = ++n - 1;
        while (i > 0 && keys[i] < keys[i-1]) {
            swap(keys[i], keys[i-1]);
            swap(childs[i], childs[i+1]);
            i--;
        }
    }
    int n = 0;
    vector<int> keys;
    vector<BTreeNode*> childs;
    static int M;
};

int BTreeNode::M = 3;

bool insertBT(BTreeNode *&T, int newKey) {
    if (T == nullptr) {
        T = new BTreeNode(newKey);
        return true;
    }
    stack<BTreeNode*> st;
    BTreeNode *x = T;
    while (x != nullptr) {
        int i = 0;
        while (i < x->n && newKey > x->keys[i]) i++;
        if (i < x->n && newKey == x->keys[i]) return false;
        st.push(x);
        x = x->childs[i];
    }

    BTreeNode *newChild = nullptr;
    while (true) {
        x = st.top(); st.pop();
        x->insertKey(newKey, newChild);
        if (x->n < BTreeNode::M) break;
        newKey = x->keys[x->n/2];
        newChild = new BTreeNode(x);
        x->keys.resize(x->n/2);
```

```
59          x->childs.resize(x->n/2+1);
60          x->n /= 2;
61          newChild->keys.erase(newChild->keys.begin(), newChild->keys.begin() + newChild->n/2+1);
62          newChild->childs.erase(newChild->childs.begin(), newChild->childs.begin() + newChild->n/2+1);
63          newChild->n -= newChild->n / 2 + 1;
64          if (st.empty()) {
65              T = new BTreeNode(newKey);
66              T->childs[0] = x;
67              T->childs[1] = newChild;
68              break;
69          }
70      }
71
72      return true;
73  }
74
75  bool deleteBT(BTreeNode *&T, int oldKey) {
76      // TODO:
77  }
78
79  void inorderBT(BTreeNode *T) {
80      if (T == nullptr) return;
81      for (int i=0;i<T->n;++i) {
82          inorderBT(T->childs[i]);
83          cout << T->keys[i] << ' ';
84      }
85      inorderBT(T->childs[T->n]);
86  }
87
88  int main() {
89      cout << " File Processing  " << endl;
90      cout << "      B-Tree" << endl;
91      cout << "==================" << endl;
92
93      int insertKeys[] = {
94          40, 11, 77, 33, 20, 90, 99, 70, 88, 80,
95          66, 10, 22, 30, 44, 55, 50, 60, 100, 28,
96          18, 9, 5, 17, 6, 3, 1, 4, 2, 7,
97          8, 73, 12, 13, 14, 16, 15, 25, 24, 28,
98          45, 49, 42, 43, 41, 47, 48, 46, 63, 68,
99          61, 62, 64, 69, 67, 65, 54, 59, 58, 51,
100         53, 57, 52, 56, 83, 81, 82, 84, 75, 89
101     };
102
103     BTreeNode *root = nullptr;
104
105     BTreeNode::M = 3;
106     cout << "M = 3" << endl;
107     for (int key : insertKeys) {
108         cout << "Insert " << key << '\t';
109         insertBT(root, key);
110         inorderBT(root);
111         cout << endl;
112     }
113
114     delete root;
```

```
115     root = nullptr;
116
117     cout << "==================" << endl;
118     BTreeNode::M = 4;
119     cout << "M = 4" << endl;
120     for (int key : insertKeys) {
121         cout << "Insert " << key << '\t';
122         insertBT(root, key);
123         inorderBT(root);
124         cout << endl;
125     }
126
127     delete root;
128
129     return 0;
130 }
```