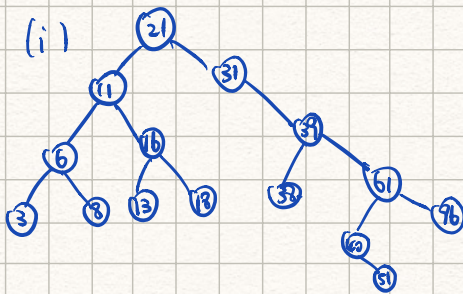


2017

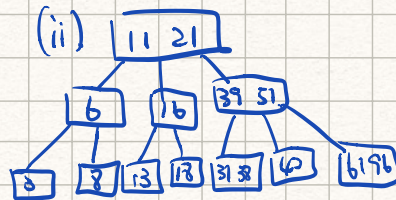
Section A.

1. a.

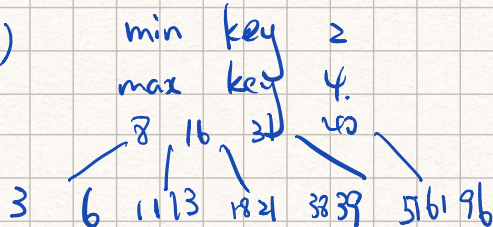


min key 1

max key 2



(iii)



b. int getPredecessor (int x) {
 if (left[x] != nil) return max(left[x]);
 else {
 y = parent[x];
 while y != nil and x == left[y]
 x = y;
 y = parent[x];
 return y;
 }

int max (int x) {
 while (right[x] != nil)
 x = right[x];
 return x

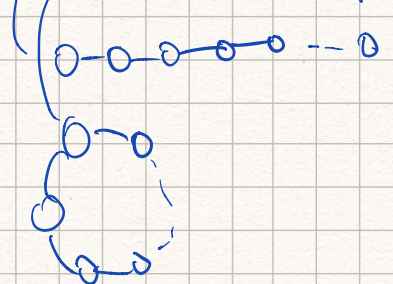
2. a.

B E S U A T N
B E S U A 1
| E 0111 010 0011 0010
| E
10 000

$$2 \times 2 + 3 \times 4 + 3 \times 2 + 2 \times 2 + 3 \times 2 + 4 + 4$$

$$= 4 + 12 + 6 + 4 + 6 + 8$$

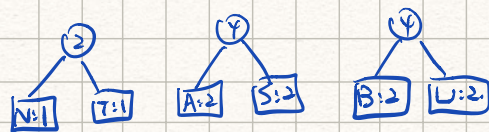
c. Max n-2 articulation points
Min 0 articulation points



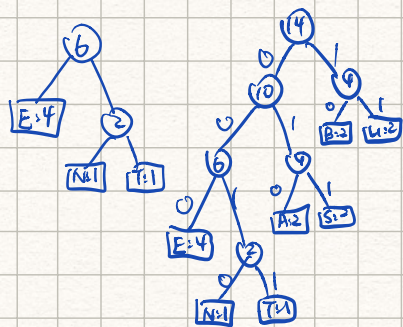
d. g is connected.
connecting n nodes
requires at least n-1 edges
such that every edge connects
a new node.

1. Each edge is a bridge.
yes

N:1 T:1 A:2 S:2 B:2 L:2 E:4



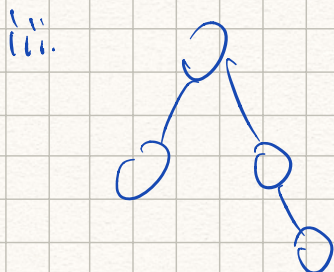
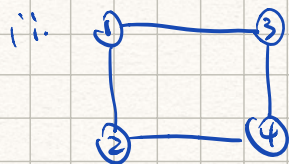
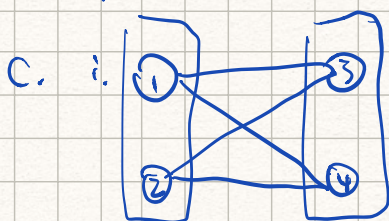
$$= 20 + 10 + 10 = 40 \text{ bes}$$



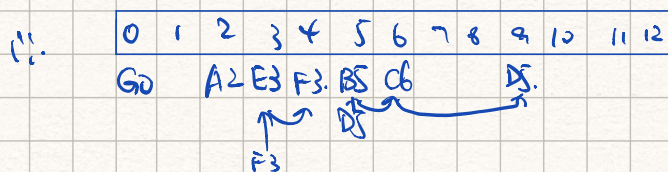
```

b. int LongestKey(l, r, x, k) {
    if (x == nil) return -INIMAX;
    int left = LongestKey(l, r, l(x), k);
    int right = LongestKey(l, r, r(x), k);
    return max(k(x), left, right);
}

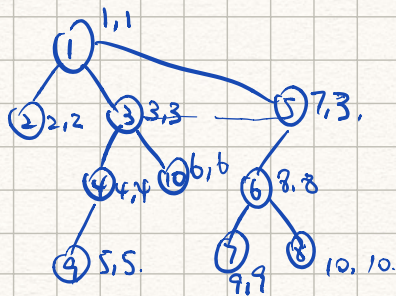
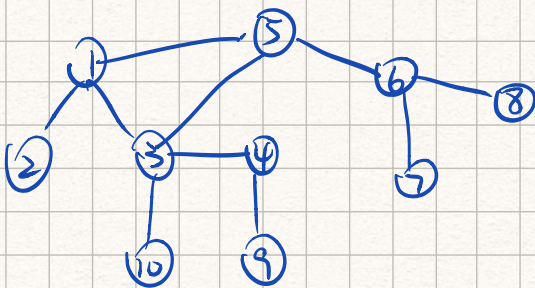
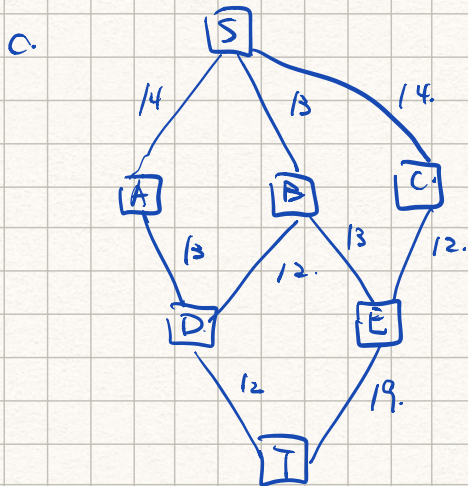
```



3. i. H $H+1^2$ $H+2^2$ $H+3^2$



b. strongly connected components
 every node x, y
 $x \rightarrow y$ $y \rightarrow x$



Node	1	2	3	4	5	6	7	8	9	10
Prem	1	2	3	4	7	8	9	10	5	6