

I hereby pledge on my honor that I will strictly adhere to academic integrity codes and the work done on this examination is solely my own and I will not receive or give any help from/to anybody or source during this examination.

1)

```

for (int j = 1; j < list.size(); j++) { →  $\Theta(n)$ 
    E c = list.get(j); →  $\Theta(n)$ 
    int i = j-1; →  $\Theta(1)$ 
    while ((i > -1) && ((list.get(i).compareTo(c)) == 1)) {
        list.set(i+1, list.get(i)); →  $\Theta(n)$ 
        i--; →  $\Theta(1)$ 
    }
    list.set(i+1, current); →  $\Theta(n)$ 
}

```

$\Theta(n^2)$

$$T_w(n) = \Theta(n^3)$$



```

Iterator<E> itr = list.iterator();
Iterator<E> itr2 = list.descendingIterator();
for (int j = 1; j < list.size(); ++j) {

```

```

    E c = itr.next();

```

```

    while (itr2.hasNext() && itr2.previous().compareTo(c) == 1) {
        itr.setNext(itr2.next());
        itr2.previous();
    }

```

```

}

```

```

itr2.setNext(c);

```

```

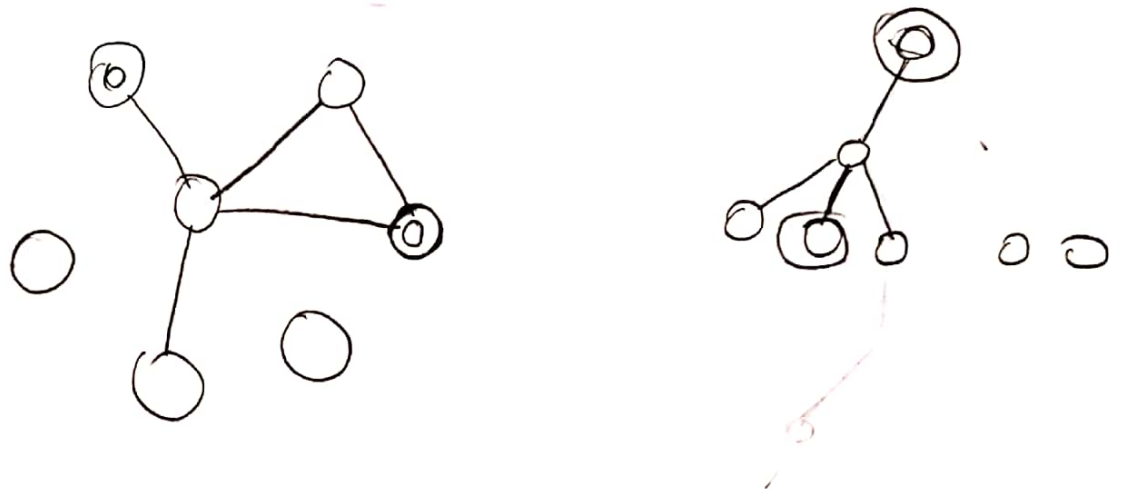
itr2 = list.descendingIterator();

```

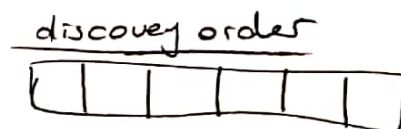
```

}

```



We are create a tree from given mode as being root. and using bread first search method. using array parent and discovery Order array we create a path from given node to k integer value. Then sum the number of nodes



T(n) Adjacency matrix;

$$\Theta(n^2)$$

Adjacency List (

$$\Theta(m.n)$$

↳ it is change for spare or dense