(Problem-3)

(1)
```
void simplevector<T>::push(T &val) {
    T *naptr;
    try { naptr = new T[arraysize+1];
    } catch (bad_alloc) {
        memerror();
    }
    for(int count=0; count<arraysize; count++) {
        naptr[count] = aptr[count];
    }
    naptr[arraysize++] = val;
    delete [] aptr;
    aptr=0;
    aptr = naptr;
}
```

Let's consider,

Operations before the for loop, O_b

Operations inside catch statement PO_s

operation inside the for loop, O_c

operation after the for loop O_a

$$O_b + \sum_{count=0}^{arraysize-1} (O_c) + O_a + PO_s ;$$

O operations which equates T(O)-clock cycle

We know, $\sum_{j=x}^{y} 1 = (y-x)+1$

Let comiden array size = n

So, $O_b + \sum_{count=0}^{n-1} (O_c) + O_a + PO_s$

$= O_b + ((n-1-0)+1) O_c + O_a + PO_s$

← f(n) - is a first-order polynomial

$= O_b + nO_c + O_a + PO_s$

$= \boxed{c'n + c}$ ⟶ it is $O(n)$

where $c' = O_c$

$c = O_b + O_a + PO_s$

(2) [With optimized simple Vector using arrays]:

```
Void SimpleVector<T>::push(T &val) {
    if (arraySize == maxSize) {
        maxSize *= 2;
        T *naptr;
    try { naptr = new T[maxSize]
    } catch (bad_alloc) {
            memError();
    }
for (int count=0; count<arraySize; count++)
    { naptr[count] = aptr[count];
    }
naptr[arraySize++] = val;
delete[] aptr;
aptr = 0;
aptr = naptr;
} else {
        aptr[arraySize++] = val;
    }
}
```

Let's consider, Operation before if statement, O_f
Operation before the for loop, O_b
Operation inside the catch statement, PO_s
Operation inside the for loop, O_c
Operations after the for loop, O_a
operations inside else statement, PO_e

from the above function, we can write,

$$O_f + O_b + PO_s + \sum_{count=0}^{arraySize-1} (O_c) + O_a + PO_e$$

O, operation which equates T(O) - clock cycles

Let's consider the array size = n

$$O_f + O_b + PO_s + \sum_{count=0}^{n-1} (O_c) + O_a + PO_e$$

We know, $\sum_{j=x}^{y} 1 = (y - x) + 1$

So, $O_f + O_b + PO_s + \underbrace{((n-1)+1)}_{\text{f(n) - is a first order polynomial}} O_c + O_a + PO_e$

$$= O_f + O_b + nO_c + PO_s + O_a + PO_e$$

$$= \boxed{c'n + e}$$

Where $c' = O_c$

$e = O_f + O_b + PO_s + O_a + PO_e$

↓

it is $O(n)$

(3) Simple vector with Linked list:

```
Void LinkedList<T>::addLst(const T &data){
    Link *end;
    temp = front;
  do{ end = temp;
      temp = temp->linkPtr;
  }while(temp != NULL);
  Link *add = new Link;
  add->data = data;
  add->linkPtr = NULL;
  end->linkPtr = add;
}
```

Let's consider, operations before the do-while loop = O_b
operations inside the do-while loop = O_d
operations after the do-while loop = O_a

$O_b + \sum_{i=0}^{n-1} O_d + O_a$; O, operations which equates T(O)- clock cycles

Here, n → is the number nodes in the list. In the do-while loop temp start from the beginning of the linked list and traverses each node until it reaches to the end

We know, $\sum_{j=x}^{y} 1 = (y-x)+1$

So, $O_b + ((n-1-0)+1)\,O_d + O_a$

$= O_b + n\,O_d + O_a \rightarrow f(n)$ is a first order polynomial

$= \boxed{c'n + c}$ where, $c' = O_d$

$c = O_b + O_a$

↑

it is $O(n)$