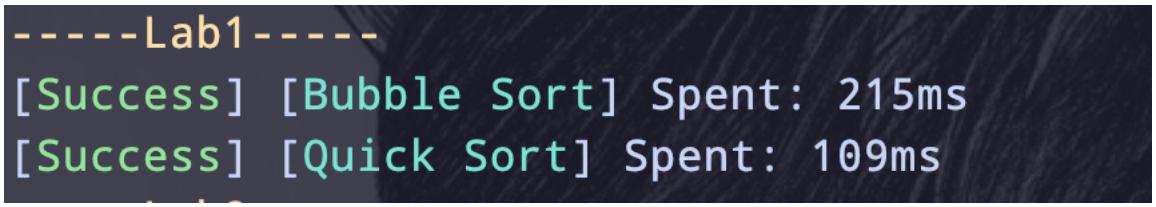
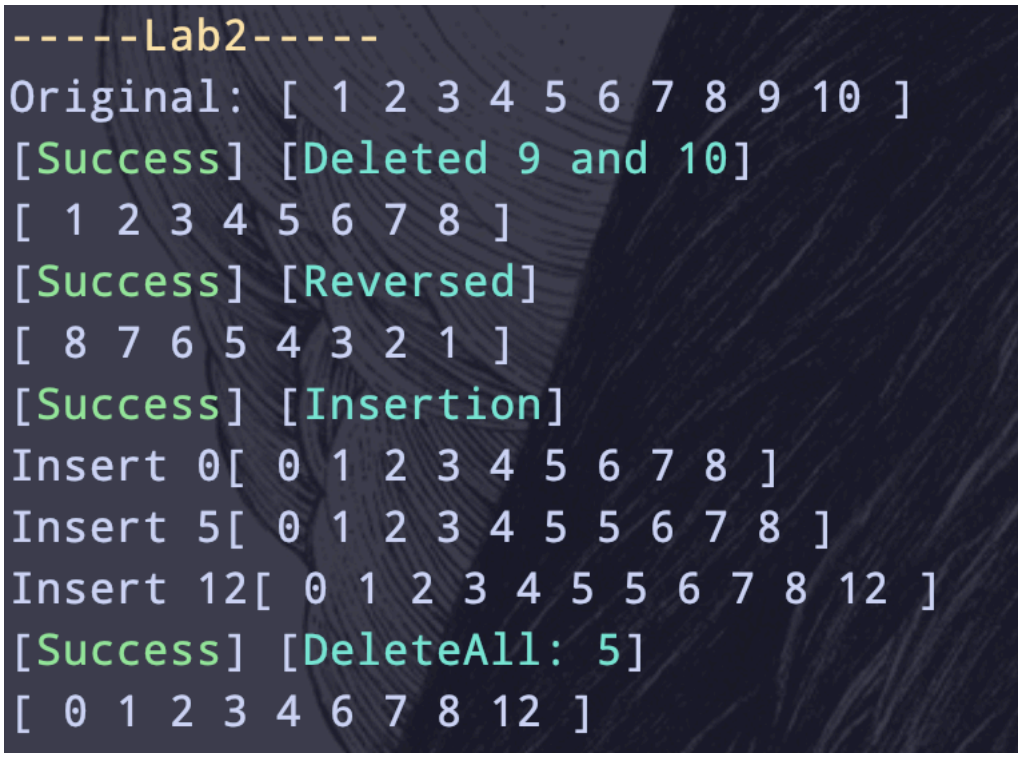

-----Lab1-----
[Success] [Bubble Sort] Spent: 215ms
[Success] [Quick Sort] Spent: 109ms


-----Lab2-----
Original: [1 2 3 4 5 6 7 8 9 10]
[Success] [Deleted 9 and 10]
[1 2 3 4 5 6 7 8]
[Success] [Reversed]
[8 7 6 5 4 3 2 1]
[Success] [Insertion]
Insert 0[0 1 2 3 4 5 6 7 8]
Insert 5[0 1 2 3 4 5 5 6 7 8]
Insert 12[0 1 2 3 4 5 5 6 7 8 12]
[Success] [DeleteAll: 5]
[0 1 2 3 4 6 7 8 12]

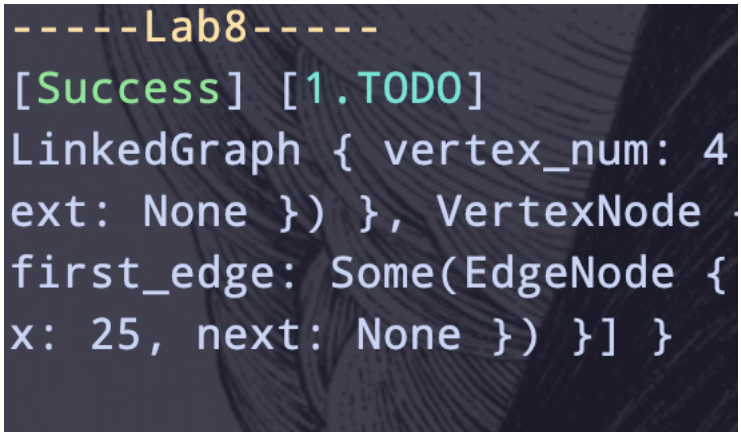
```
-----Lab3-----
Original: [ 1 1 2 3 3 4 5 5 10 20 20 ]
[Success] [Delete 1 5 3]
[ 1 2 3 4 5 10 20 20 ]
[Success] [Reversed]
[ 20 20 10 5 4 3 2 1 ]
[Success] [Delex 20 10 3 1]
[ 5 4 2 ]
[Success] [Merge 1 6 and ascend]
[ 1 2 4 5 ]
[Success] [Merge 1 6 and Descend]
[ 5 4 2 1 1 ]
[Success] [Intersection]
: of [ 1 6 3 2 ] and [ 1 6 3 2 ] is [ 2 3 6 1 ]
[Success] [Search: last 1, 2]
[ 2 3 6 1 ] is 1 6
[Success] [Relocate odd number]
[ 1 9 7 3 3 3 12 4 12 0 ]
```

```
-----Lab4-----
[Success] [DEC转换HEX]
十进制255对应的十六进制数是：FF
十进制510对应的十六进制数是：1FE
[Success] [算数表达式求值：{'3','+','4','*','5'}]
表达式结果为：35
[Success] [子字符串：test_string, i=5,len=6]
string
[Success] [删除子字符串：test_string, i=5,len=6]
test_
[Success] [查找第一次出现的位置：test_, t=es]
1
[Success] [朴素匹配]
第一次的位置：testes_es_es_, t=es
1
所有位置：testes_es_es_, t=es
10741
[Success] [KMP]
7
```

```
-----Lab5-----
[Success] [1.求a[left..right]最大数]
10
[Success] [2.调整[left..right]中的奇数到左边 偶数到右边]
Before: Array[ 1 2 3 10 5 6 7 8 9 9 9 9 9 ]
After: Array[ 9 9 9 9 1 3 5 7 9 10 8 6 2 ]
[Success] [3.1.冒泡升序排序[left..right]]
Array[ 1 2 3 5 6 7 8 9 9 9 9 9 10 ]
[Success] [3.2.二分查找[left..right]中的key[8]]
6原链表: [ 7 2 3 4 5 6 7 7 4 4 10 7 ]
[Success] [4.1.查找链表最大值]
10
[Success] [4.2.删除链表所有x]
[ 2 3 4 5 6 4 4 10 ]
[Success] [4.2.删除链表直到只剩一个x]
[ 7 2 3 4 5 6 4 4 10 ]
```

```
-----Lab6-----
现有二叉树tree1:A(B,C(E,F(H,I),G),D)
        tree2:AL(B,C(D,E),F(G,H(I,J)))
[Success] [1.层次遍历tree1]
A,D,C,B,G,F,E,I,H,
[Success] [2.非递归tree1前序遍历]
A,D,C,G,F,I,H,E,B,
[Success] [3.非递归后续遍历tree1]
D,G,I,H,F,E,C,B,A,
[Success] [4.判断两颗树(tree1,tree2)是否等价]
0
```

```
-----Lab7-----
示例表达式: A(B(S,D),C(E,F))
[Success] [1.前序非递归遍历]
A B S D C E F
[Success] [2.层次非递归遍历]
A B C S D E F
[Success] [3.前序遍历最后一个节点]
F
[Success] [4.后序遍历第一个节点]
S
[Success] [5.左右子女互换]
A C B F E D S
[Success] [6.表达式建树]
Expression:A(B(S,D),C(E,F))
层次非递归遍历: A B C S D E F
```


-----Lab8-----
[Success] [1.TODO]
LinkedGraph { vertex_num: 4
ext: None }) }, VertexNode
first_edge: Some(EdgeNode {
x: 25, next: None }) }] }

```
-----Lab9-----
1.基于数组顺序查找:
[Unsuccess] [Size: 10000] Spent: 27us
[Unsuccess] [Size: 50000] Spent: 101us
[Unsuccess] [Size: 100000] Spent: 180us
[Success] [Size: 200000] Spent: 280us
[Success] [Size: 300000] Spent: 275us
[Success] [Size: 400000] Spent: 276us
[Success] [Size: 500000] Spent: 283us
2.基于单链表顺序查找:
[Success] [Size: 10000] Spent: 15us
[Success] [Size: 50000] Spent: 8us
[Success] [Size: 100000] Spent: 8us
[Success] [Size: 200000] Spent: 7us
[Success] [Size: 300000] Spent: 7us
[Success] [Size: 400000] Spent: 8us
[Success] [Size: 500000] Spent: 8us
```

```
3.基于数组非递归二分查找:
[Success] [Size: 10000] Spent: 1us
[Success] [Size: 50000] Spent: 1us
[Success] [Size: 100000] Spent: 678ns
[Success] [Size: 200000] Spent: 742ns
[Success] [Size: 300000] Spent: 1us
[Success] [Size: 400000] Spent: 801ns
[Success] [Size: 500000] Spent: 1us

4.基于数组递归二分查找:
[Success] [Size: 10000] Spent: 715ns
[Success] [Size: 50000] Spent: 680ns
[Success] [Size: 100000] Spent: 627ns
[Success] [Size: 200000] Spent: 689ns
[Success] [Size: 300000] Spent: 641ns
[Success] [Size: 400000] Spent: 650ns
[Success] [Size: 500000] Spent: 2us
```

5.为数组a的数据建立二叉排序树并查找:

```
[Success] [Size: 10000] Spent: 2us
[Success] [Size: 50000] Spent: 520ns
[Success] [Size: 100000] Spent: 400ns
[Success] [Size: 200000] Spent: 369ns
[Success] [Size: 300000] Spent: 305ns
[Success] [Size: 400000] Spent: 297ns
[Success] [Size: 500000] Spent: 266ns
```

```
6.递归查找输出二叉排序树中关键字不小于k(499990)的节点值:
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 10000] Spent: 115ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 50000] Spent: 104ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 100000] Spent: 106ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 200000] Spent: 112ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 300000] Spent: 113ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 400000] Spent: 109ms
{ 500000 499999 499998 499997 499996 499995 499994 499993 499992 499991 }
[Success] [Size: 500000] Spent: 112ms
```

7.求二叉排序树指定节点层次，若树为空返回-1 若节点不在树中返回0:
{ 1999 } 返回: 1
[Success] [Size: 10000] Spent: 107ms
{ 1999 } 返回: 1
[Success] [Size: 50000] Spent: 116ms
{ 1999 } 返回: 1
[Success] [Size: 100000] Spent: 111ms
{ 1999 } 返回: 1
[Success] [Size: 200000] Spent: 108ms
{ 1999 } 返回: 1
[Success] [Size: 300000] Spent: 111ms
{ 1999 } 返回: 1
[Success] [Size: 400000] Spent: 107ms
{ 1999 } 返回: 1
[Success] [Size: 500000] Spent: 109ms

```
8.判断是否从查找序列搜索的k关键字:
测试序列: { 12 213  } 0
[Success] [Size: 10000] Spent: 13us
测试序列: { 12 213  } 0
[Success] [Size: 50000] Spent: 3us
测试序列: { 12 213  } 0
[Success] [Size: 100000] Spent: 3us
测试序列: { 12 213  } 0
[Success] [Size: 200000] Spent: 3us
测试序列: { 12 213  } 0
[Success] [Size: 300000] Spent: 3us
测试序列: { 12 213  } 0
[Success] [Size: 400000] Spent: 3us
测试序列: { 12 213  } 0
[Success] [Size: 500000] Spent: 3us
```

```
-----Lab10-----
[Success] [Radix Sort Size: 10000] Spent: 195ms
[Success] [Radix Sort Size: 20000] Spent: 281ms
[Success] [Radix Sort Size: 50000] Spent: 274ms
[Success] [Radix Sort Size: 100000] Spent: 295ms
[Success] [Radix Sort Size: 300000] Spent: 302ms
[Success] [Radix Sort Size: 400000] Spent: 303ms
[Success] [Radix Sort Size: 500000] Spent: 264ms
[Success] [Quick Sort Size: 10000] Spent: 1ms
[Success] [Quick Sort Size: 20000] Spent: 2ms
[Success] [Quick Sort Size: 50000] Spent: 7ms
[Success] [Quick Sort Size: 100000] Spent: 13ms
[Success] [Quick Sort Size: 300000] Spent: 44ms
[Success] [Quick Sort Size: 400000] Spent: 67ms
[Success] [Quick Sort Size: 500000] Spent: 77ms
[Success] [Heap Sort Size: 10000] Spent: 1ms
[Success] [Heap Sort Size: 20000] Spent: 2ms
[Success] [Heap Sort Size: 50000] Spent: 9ms
[Success] [Heap Sort Size: 100000] Spent: 18ms
[Success] [Heap Sort Size: 300000] Spent: 57ms
[Success] [Heap Sort Size: 400000] Spent: 82ms
[Success] [Heap Sort Size: 500000] Spent: 102ms
[Success] [Merge Sort Size: 10000] Spent: 2ms
[Success] [Merge Sort Size: 20000] Spent: 5ms
[Success] [Merge Sort Size: 50000] Spent: 14ms
[Success] [Merge Sort Size: 100000] Spent: 29ms
```

```
[Success] [Merge Sort Size: 300000] Spent: 89ms
[Success] [Merge Sort Size: 400000] Spent: 125ms
[Success] [Merge Sort Size: 500000] Spent: 150ms
[Success] [Shell Sort Size: 10000] Spent: 505us
[Success] [Shell Sort Size: 20000] Spent: 1ms
[Success] [Shell Sort Size: 50000] Spent: 2ms
[Success] [Shell Sort Size: 100000] Spent: 5ms
[Success] [Shell Sort Size: 300000] Spent: 17ms
[Success] [Shell Sort Size: 400000] Spent: 22ms
[Success] [Shell Sort Size: 500000] Spent: 30ms
[Success] [Binary Insertion Sort Size: 10000] Spent: 1ms
[Success] [Binary Insertion Sort Size: 20000] Spent: 2ms
[Success] [Binary Insertion Sort Size: 50000] Spent: 6ms
[Success] [Binary Insertion Sort Size: 100000] Spent: 15ms
[Success] [Binary Insertion Sort Size: 300000] Spent: 49ms
[Success] [Binary Insertion Sort Size: 400000] Spent: 69ms
[Success] [Binary Insertion Sort Size: 500000] Spent: 87ms
[Success] [Select Sort Size: 10000] Spent: 94ms
[Success] [Select Sort Size: 20000] Spent: 374ms
[Success] [Select Sort Size: 50000] Spent: 2s
[Success] [Select Sort Size: 100000] Spent: 9s
[Success] [Select Sort Size: 300000] Spent: 91s
[Success] [Select Sort Size: 400000] Spent: 166s
[Success] [Select Sort Size: 500000] Spent: 236s
[Success] [Insertion Sort Size: 10000] Spent: 44us
[Success] [Insertion Sort Size: 20000] Spent: 70us
[Success] [Insertion Sort Size: 50000] Spent: 216us
[Success] [Insertion Sort Size: 100000] Spent: 345us
```

```
[Success] [Insertion Sort Size: 300000] Spent: 1ms
[Success] [Insertion Sort Size: 400000] Spent: 1ms
[Success] [Insertion Sort Size: 500000] Spent: 1ms
[Success] [Bubble Sort Size: 10000] Spent: 175ms
[Success] [Bubble Sort Size: 20000] Spent: 611ms
[Success] [Bubble Sort Size: 50000] Spent: 4s
[Success] [Bubble Sort Size: 100000] Spent: 15s
[Success] [Bubble Sort Size: 300000] Spent: 146s
[Success] [Bubble Sort Size: 400000] Spent: 200s
[Success] [Bubble Sort Size: 500000] Spent: 311s
```