## 20190408 左士海-作业 6

1. 画出下列广义表的存储结构,写出其长度、深度以及表头和表尾。

```
A=(a,b,c)
          B = (a, (b, (c)), d)
          C = ((a, b), (c, d))
          D = (a, (b, (), c), ((d), e))
          E = ((((a), b)), (((), (d)), (ef)))
                                                        D = (a, (b, (), c), ((d), e))
  A = (a, b, c)
 [1]_{1}
        \overline{\phantom{a}}
              length = 3, depth = 1
                                                                     length = 3, depth = 3
                                                         \square
 Oa
                      <del>)></del>(0 | c | ^
                                                         0 a
                                                                      [1]
                                                                                 <del>></del>[1]
                                                                                           →(O c ^)
                                                                      0 6
  B = (a, (b, c), d)
                                                                                    (0 d ^)
                                                                             \wedge
                length = 3, depth = 3
                                                                             000
  (0 a
                                                         E = ((((a), b)), (((), (d)), (e, f)))
              (0 b
                                                         (1)_1
                           (O c ^
                                                                                    length = 2, depth = 4
                                                     head[1
C = ((a, b), (c, d))
                                                         (1)
1 1 ^ length = 2, depth = 2
                                                                                       >111^
[1]
                                                         (1)
                                                                                   0 0 0
```

(0 a 1

2. 试编写判别两个广义表是否相等的算法。

(0 c )>(0 d ^)

(0 a -> 0 b ^

对于某个节点,其存在三种类型: ATOM、LIST 和 NULL。使用递归式 的方法进行比较,当前节点匹配成功,则进入其后继的比较。

```
template<class T>
bool GList<T>::isSame(GListNode<T> *p, GListNode<T> *q) {
     if (p == nullptr) {
    return q == nullptr;
} else if (p->type == ATOM) {
    return q != nullptr && q->type == ATOM && q->data == p->data && isSame(p->next, q->next);
                    q != nullptr && q->type == LIST && isSame(p->sublist, q->sublist) && isSame(p->next, q->next);
```

3. 试编写一个算法,删除广义表中所有元素值为 x 的原子结点。

类似于广义表的遍历算法,如匹配到当前原子值为 x,则进行节点的删除,以及指针的更新。

```
template<class T>
GListNode<T> * GList<T>::Delete(GListNode<T> *p, T x) {
    if(p == NULL)
        return NULL;
    GListNode<T> * q = Delete(p->next, x);
    if(p->type == ATOM) {
        if(p->data == x) {
            delete []p;
            return q;
        }
    }
    p->next = q;
    if(p->type == LIST) {
        p->sublist = Delete(p->sublist, x);
    }
    return p;
}
```