Start of powerSet homework problem assuming/using Lists.

```
Algorithm powerSet(L)
    res := new List
    if L.size() = 0 then
        res.insertLast(L)
        return res
    e := L.remove(L.first())
    pow := powerSet(L)
    p := pow.first()
    addNextPair(p.element(), e, res)
    while !pow.isLast(p) do
        p := pow.after(p)
        addNextPair(p.element(), e, res)
    return res
Algorithm addNextPair(subset, e, res)
    subsetClone := cloneInsert(subset, e)
    res.insertLast(subset)
    res.insertLast(subsetClone)

Algorithm cloneInsert(subset, e)
    subsetClone := new List
    subsetClone.insertFirst(e)
    if subset.isEmpty() then
        return subsetClone
    p := subset.first()
    subsetClone.insertLast(p.element())
    while !subset.isLast(p) do
        p := subset.after(p)
        subsetClone.insertLast(p.element())
```

Start of powerSet homework problem assuming Sequences.

```
Algorithm powerSet(S)
    res := new Sequence
    if S.size() = 0 then
        res.insertLast(S)
        return res
    e := S.remove(S.first())
    pow := powerSet(S)
    for i := 0 to pow.size()-1 do
        subset := pow.elemAtRank(i))
        subsetClone := cloneInsert(subset, e)
        res.insertLast(subset)
        res.insertLast(subsetClone)
    return res

Algorithm cloneInsert(subset, e)
```