Merge Sort

```
mergesort(m){
    if (m.length() ≤ 1)    return m;
    list left, right, result;
    int middle = m.length()/2;
    for (int i=0, i≤middle, i++)
        left.push(m[i]);
    for (int i=middle+1; i<m.length(); i++)
        right.push(m[i]);

    left = mergesort(left);
    right = mergesort(right);
    result = merge(left, right);
    return result;
}
```

```
merge(l, r){
    list result;
    int lx=0, rx=0;
    while(lx<l.length() || rx<r.length()){
        if(lx<l.length() && rx<r.length()){
            if(l[lx]≤r[rx]){
                result.push(l[lx]);
                lx++;
            } else {
                result.push(r[rx]);
                rx++;
            }
        } else if(lx<l.length()){
            result.push(l[lx]);
            lx++;
        } else if(rx<r.length()){
            result.push(r[rx]);
            rx++;
        }
    }
    return result;
}
```