2) **Practice with recursion, i.e., the concept of reducing (an instance of) a problem to a smaller instance of the same problem.**

(a) **sorting a list using selection sort**

```
selection_sort(A,start,end){
if(start==end)
        return;
min=infinity;
min_pos=start;
min_pos=smallest(A,infinity,start);
swap(A[start],A[min_pos]);
selection_sort(A,start+1,end);
}

int smallest(A,small,pos)
{
if(pos==A.length)
        return small;
if(A[pos]<A[small]){
        small=pos;
}
return smallest(A,small,pos+1);
}
```

(b) **determining whether an item x appears in a list L**

```
search(A,element){
if(A==null){
return false;
}
if(first[A]==element)
return true
else
return search(rest[A],element);
}
```

(c) **finding the smallest number in a list of numbers L**

```
smallest(A,small)
{
if(A==null)
return small;
if(first(A)<small){
small=first(A);
}
return smallest(rest(A),small);
}
smallest(A,∞);   //function call
```