



Programmierung: Musterlösung zum 4. Übungsblatt

Prof. Gert Smolka und Thorsten Brunklaus

Aufgabe 4.1: Last (3)

```
fun last (x::nil) = x
  | last (_::xr)  = last xr
  | last _       = raise Empty
```

Aufgabe 4.2: Enum (3)

```
fun enum (m,n) = if m>n then nil else m :: enum(m+1,n)
```

Aufgabe 4.3: Nth, Take, Drop (4+4+4)

```
fun nth(xs,n) = if n<0 orelse null xs
  then raise Subscript
  else if n=0 then hd xs
  else nth(tl xs, n-1)

fun take(xs,n) = if n=0 then nil
  else if n<0 orelse null xs
  then raise Subscript
  else hd xs :: take(tl xs, n-1)

fun drop(xs,n) = if n=0 then xs
  else if n<0 orelse null xs
  then raise Subscript
  else drop(tl xs, n-1)
```

Aufgabe 4.4: Member (3+3+3)

```
fun member x nil      = false
  | member x (y::yr) = x=y orelse member x yr

fun member x = List.exists (fn y => x=y)

fun member x = foldl (fn (y,b) => x=y orelse b) false
```

Aufgabe 4.5: Partition (4+4)

```
fun partition x nil      = (nil, nil)
  | partition x (y::yr) = let
    val (us,vs) = partition x yr
  in
    if y<x then (y::us, vs)
    else (us, y::vs)
  end
```

```

fun partition x = foldl (fn (y, (us,vs)) =>
                        if y<x then (y::us, vs)
                        else (us, y::vs))
                        (nil,nil)

```

Aufgabe 4.6: Quicksort (8)

```

fun qsort nil      = nil
  | qsort (x::xs) = let
                        val (us,vs) = partition x xs
                      in
                        qsort us @ [x] @ qsort vs
                      end

```

Aufgabe 4.7: Polymorphes Mergesort (7)

```

fun pmsort order =
  let
    fun split xs = foldl (fn (x, (ys, zs)) => (zs, x::ys))
                        (nil, nil) xs

    fun merge (nil , ys ) = ys
      | merge (xs , nil ) = xs
      | merge (x::xr, y::yr) = if order(x,y) then x::merge(xr,y::yr)
                                else y::merge(x::xr,yr)

    fun sort nil      = nil
      | sort (x::nil) = x::nil
      | sort xs       = let
                            val (ys,zs) = split xs
                          in
                            merge(sort ys, sort zs)
                          end
  in
    sort
  end

```