

Tutorial 5

Exercise 1

- **merge**(A, B) =
 $i := 1; j := 1; k := 1$
while $i \leq n$ or $j \leq m$ **do**
 if $i > n$ **then**
 $C[k] := B[j]$
 $k := k + 1; j := j + 1$
 else if $j > n$ **then**
 $C[k] := A[i]$
 $k := k + 1; i := i + 1$
 else if $A[i] \leq B[j]$ **then**
 $C[k] := A[i]$
 $k := k + 1; i := i + 1$
 else
 $C[k] := B[j]$
 $k := k + 1; j := j + 1$
 end if
end while
return C
- $O(n + m)$

Here is another solution (inspired by some of the groups):

```

merge( $A, B$ ) =
 $i := 1; j := 1$ 
for  $k := 1$  to  $(m + n)$  do
  if  $i > n$  then
     $C[k] := B[j]; j := j + 1$ 
  else if  $j > n$  then
     $C[k] := A[i]; i := i + 1$ 
  else if  $A[i] \leq B[j]$  then
     $C[k] := A[i]; i := i + 1$ 
  else
     $C[k] := B[j]; j := j + 1$ 
  end if
end for

```

Exercise 2

- **sum**($s : SET$): $int =$
 $x := 0$
 $tmp.make$
 while $not\ s.empty$ **do**
 $elem := s.delete_any$
 $x := x + elem$
 $tmp.insert(elem)$
 end while
 while $not\ tmp.empty$ **do**
 $s.insert(tmp.delete_any)$
 end while
 return x
- Yes - $O(n)$

Exercise 3

- **intersection**($s_1 : SET, s_2 : SET$): $SET =$
 $res.make, tmp.make$
 while $not\ s_1.empty$ **do**
 $elem := s_1.delete_any$
 $tmp.insert(elem)$
 if $member(elem, s_2)$ **then**
 $res.insert(elem)$
 end if
 end while
 while $not\ tmp.empty$ **do**
 $s_1.insert(tmp.delete_any)$
 end while
 return res
- $O(|s_1| \cdot |s_2|)$