Sequence Algorithms

Algorithm 1: Searching a sequence for the maximum value

Input: Sequence *S*, length *n*
Output: *max*, maximum value

(a) Using a while loop

```
procedure findMax(S,n)
  {
    max=s_1
    i=1
    while (i<n)
      {
        if (s_i>max)
          max=s_i
        i=i+1
      }
    return(max)
  } findMax
```

(b) Using a for loop

```
procedure findMax(S,n)
  {
    max=s_1
    for i=2 to n
      if (s_i>max)
        max=s_i
    return(max)
  } findMax
```

Algorithm 2: Finding the first occurrence of the largest member

Input: Sequence *S*, length *n*
Output: *max*, maximum value

(a) Using a while loop

```
procedure findFirstMax(S,n)
  {
    max=s_1
    index=1
    i=2
    while (i<=n)
```

```
    {
      if (s_i>max)
      {
        max=s_i
        index=i
      }
      i=i+1
    }
  return(index)
} findFIrstMax
```

(a) Using a for loop

```
procedure findFirstMax(S,n)
  {
    max=s_1
    index=1
    for i=2 to n
      if (s_i>max)
        {
          max=s_i
          index=i
        }
    return(index)
    } findFirstMax
```