Here we demonstrate how algorithms or pseudocode can be typeset using the algorithm environment provided by the algorithm2e package.

You should not load the algorithm, algorithm, algorithmic packages if you have already loaded algorithm2e.

Note that the command and argument syntax provided by algorithm2e are very different from those provided by algorithm2e. It is important to know clearly which package that you are using, and then accordingly write the relevant commands with the correct syntax.

```
\begin{array}{l} i \leftarrow 10; \\ \textbf{if } i \geq 5 \textbf{ then} \\ \mid i \leftarrow i-1; \\ \textbf{else} \\ \mid \textbf{if } i \leq 3 \textbf{ then} \\ \mid i \leftarrow i+2; \\ \textbf{end} \\ \textbf{end} \end{array}
```

Every line in your source code **must** end with \; otherwise your algorithm will continue on the same line of text in the output. Only lines with a macro beginning a block should not end with \;.

The above algorithm example is uncaptioned. If you need a caption for your algorithm, use \caption{...} inside the algorithm environment. You can then use \label{...} after the \caption so that the algorithm number can be cross-referenced, e.g. Algorithm 1 and ??.

By default, the plain algorithm style is used. But if you prefer lines around the algorithm and caption, you can add the ruled package option when loading algorithm2e, or write \RestyleAlgo{ruled} in your document.

## **Algorithm 1:** An algorithm with caption

```
Data: n > 0
Result: y = x^n
y \leftarrow 1;
X \leftarrow x;
N \leftarrow n;
while N \neq 0 do
    if N is even then
         X \leftarrow X \times X;
        N \leftarrow \frac{N}{2};
                                                                           /* This is a comment */
    else
         if N is odd then
             y \leftarrow y \times X;
             N \leftarrow N-1;
         end
    end
end
```

## **Algorithm 2:** Bubble Sort Algorithm

```
Data: An array A of n elements

Result: The array A sorted in non-decreasing order for i \leftarrow 0 to n-1 do

| for j \leftarrow 0 to n-i-1 do
| if A[j] > A[j+1] then
| swap A[j] and A[j+1];
| end
| end
| end
```

## **Algorithm 3:** Binary Search Algorithm

```
Data: An array A of n elements sorted in non-decreasing order, and a search key x
Result: The index of x in A, or -1 if x is not found
low \leftarrow 0;
high \leftarrow n-1;
while low \leq high \ do
   mid \leftarrow \left\lfloor \frac{low + high}{2} \right\rfloor;
                                                         /* Compute the midpoint */
   if A[mid] = x then
    return mid;
                                                         /* Found x at index mid */
   end
   if A[mid] < x then
    low \leftarrow mid + 1;
                                                /* x must be in the right half */
   end
   else
    high \leftarrow mid - 1;
                                                 /* x must be in the left half */
   end
end
return -1;
                                                        /* x is not in the array */
```

The algorithm2e package also provides a \listofalgorithms command that works like \listoffigures, but for captioned algorithms:

## List of Algorithms