

Algorithm Design and Analysis, Colby College CS 376
(Prof. Eric Aaron)

L^AT_EX Templates and Samples for Algorithms and Specifications

- These are intended as examples you might follow when presenting algorithms and specifications in CS376. (Thank you for working with L^AT_EX this semester!) You do not need to follow these examples—indeed, searching online will probably turn up other, nicer looking ways to present material like this—but these are relatively simple options that can get you started, and they should be sufficient for CS376.

1 Using the verbatim environment

The `verbatim` environment typesets text in `teletype` font, and it preserves exactly the spacing of the material typed in the `tex` file. For example, a Bubble Sort algorithm:

```
// Input: Array A of real numbers
BubbleSort(A)
for i = 1 to A.length - 1
  for j = A.length downto i+1
    if A[j] < A[j-1]
      swap A[j] with A[j-1]
```

2 Using a tabular environment

The `tabular` environment is a very useful thing! It uses the `&` to set positions of “tab stops,” and we could also manually set spacing for indentation (using `hspace`). For example, here’s one way it could be used to present a Selection Sort algorithm:

```
SELECTIONSORT(A[1...n])
  for i = 1 to length[A] - 1
    min = i
    for j = i + 1 to length[A]
      if A[j] < A[min]
        min = j
    // the next 3 lines swap A[i] and A[min], using a temporary variable
    temp = A[i]
    A[i] = A[min]
    A[min] = temp
```

3 Using the quote environment

The `quote` environment both left-indent and right-indent its contents, which can be useful when presenting specifications or algorithms, to make them more easily readable when among other text. For example, here's a specification for a function to reverse the elements of a list:

```
# Input: List  $L = [a_0, a_1, \dots, a_n]$ 
# Output: List  $L' = [a_n, \dots, a_1, a_0]$  with the same elements as in  $L$ 
#           but in reverse order
```

That specification didn't use the `verbatim` environment to get teletype font—in part, this was so we could have some symbols in math mode along with the teletype font. (In `verbatim` mode, a `$` symbol would be printed as `$`, not taken as the start of math mode! Also, notice the use of `hspace` to indent the second line.) Here's another specification, not quite the same as the previous one, using the `verbatim` environment:

```
# Input: LList L
# Output: LList L' that contains the elements of L
#           in reverse order
```

Here are some other examples, in case they're useful.

```
# Input: Number  $x$  and sorted LList  $L = [a_0, a_1, \dots, a_n]$ ,
#           where  $a_0 \leq a_1 \leq \dots \leq a_n$ 
# Output: List  $L' = [b_0, b_1, \dots, b_{n+1}]$  containing input  $x$  and the
#            $n + 1$  elements of  $L$ , in sorted order
#            $b_0 \leq b_1 \leq \dots \leq b_{n+1}$ 
```

```
# Input: IntBinTree T
# Output: The sum of all of the integers in tree T
```

If you have any questions about the $\text{\LaTeX}$ contents of this document, I encourage you to do a short search online for answers and then come to me to talk about what you found—I'm happy to help with your questions, but looking for answers yourself is a very helpful part of the learning process for $\text{\LaTeX}$!