CS3000: Algorithms & Data — Summer '23 — Laney Strange

Mergesort Algorithm

We went over the divide-and-conquer Mergesort algorithm in class; this handout is so you can see the function typeset in the CLRS pseudocode style. The algorithm *divides* by making the input array smaller and smaller and smaller until it's trivially easy to sort (sorting the trivially easy is the *conquer* step). Then it merges together the smaller sorted subarrays – that's the *combine* step.

```
MERGESORT($A, p, q, r$)
if $p < r$
    $q = \lfloor (p + r)/2 \rfloor$
    MERGESORT($A, p, q$)
    MERGESORT($A, q + 1, r$)
    MERGE($A, p, q, r$)
```

```
MERGE($A, p, q, r$)
$n_L = q - p + 1$
$n_R = r - q$
let $L[1 : n_L + 1]$ and $R[1 : n_R + 1]$ be new arrays
for $i = 1$ to $n_L$
    $L[i] = A[p + i - 1]$
for $j = 1$ to $n_R$
    $R[j] = A[q + j]$
$L[n_L + 1] = \infty$
$R[n_R + 1] = \infty$
$i = 1$
$j = 1$
$k = p$
while $k \le r$
    if $L[i] \le R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else
        $A[k] = R[j]$
        $j = j + 1$
    $k = k + 1$
```

Here's how we typeset the above functions in CLRS style:

```
\begin{codebox}
\Procname{$\proc{Mergesort}(A, p, r)$}
\li \If $p \ge r$
\Then
\li \Return
```

```
\li $q = \lfloor(p+r)/2\rfloor$
\li $\proc{Mergesort}(A, p, q)$
\li $\proc{Mergesort}(A, q+1, r)$
\li $\proc{Merge}(A, p, q, r)$
\end{codebox}

\medskip


\begin{codebox}
\Procname{$\proc{Merge}(A, p, q, r)$}
\li $n_L = q - p + 1$
\li $n_R = r-q$
\li let $L[1:n_L + 1]$ and  $R[1:n_R + 1]$ be new arrays
\li \For $i \gets 1 \To n_L$
\Do
\li $L[i] = A[p+i-1]$
\End
\li \For $j \gets 1 \To n_R$
\Do
\li $R[j] = A[q+j]$
\End
\li $L[n_L + 1] = \infty$
\li $R[n_R + 1] = \infty$
\li $i = 1$
\li $j = 1$
\li $k = p$
\li \While $k \le r$
\Do
\li \If $L[i] \le R[j]$
\Then
\li $A[k] = L[i]$
\li $i = i + 1$
\li \Else
\li $A[k] = R[j]$
\li $j = j + 1$
\End
\li $k = k + 1$
\End
\end{codebox}
```