

210 ($n \times \log n$ sort) Write a program to sort a list. Execution time should be at most $n \times \log n$ where n is the length of the list.

After trying the question, scroll down to the solution.

§ There are many ways to sort in time $n \times \log n$. I'll do merge sort. Let the list variable be L . Let i , j , and mid be natural variables, and let M be a list variable with the same type and length as L . Define specifications $mergesort$, $ms\ h\ k$, $perm\ A\ B$, and mm as follows.

$$mergesort = ms\ 0\ (\#L)$$

$$ms\ h\ k = (\forall a, b: h, ..k. a \leq b \Rightarrow L'a \leq L'b) \wedge perm\ (L'[h;..k])\ (L[h;..k]) \\ \wedge L'[0;..h] = L[0;..h] \wedge L'[k;..\#L] = L[k;..\#L] \\ \wedge t' = t + (k-h) \times \log(k-h)$$

$$perm\ A\ B = \forall x. \wp(\{i: 0,..\#A. A\ i = x\}) = \wp(\{i: 0,..\#B. B\ i = x\})$$

$$mm = (\text{merge } M[i;..mid] \text{ and } M[j;..k] \text{ to } L[i+j-mid;..k]) \\ \wedge t' = t + mid - i + k - j$$

Here are the refinements.

$$mergesort \Leftarrow ms\ 0\ (\#L)$$

$$ms\ h\ k \Leftarrow \text{if } k-h < 2 \text{ then } ok \\ \text{else } t := t+1. \\ ms\ h\ (\text{div}(h+k)\ 2). ms\ (\text{div}(h+k)\ 2)\ k. \\ M := M[0;..h]; L[h;..k]; M[k;..\#M]. \\ i := h. mid := \text{div}(h+k)\ 2. j := mid. mm\ \mathbf{fi}$$

$$mm \Leftarrow \text{if } i=mid \text{ then } L := L[0;..j]; M[j;..k]; L[k;..\#L] \\ \text{else if } j=k \text{ then } L := L[0;..i+k-mid]; M[i;..mid]; L[k;..\#L] \\ \text{else if } M\ i \leq M\ j \text{ then } L := i+j-mid \rightarrow M\ i \mid L. i := i+1. t := t+1. mm \\ \text{else } L := i+j-mid \rightarrow M\ j \mid L. j := j+1. t := t+1. mm\ \mathbf{fi}\ \mathbf{fi}\ \mathbf{fi}$$

If we make variables i , j , and mid local to ms (see Section 5.0), then we can make the two calls to ms concurrent (see Chapter 8).

This version of *mergesort* is top-down: sort the halves, then merge them. There is also bottom-up mergesort: first sort pairs, then merge pairs of pairs, then merge pairs of fours, and so on. And there are many other $n \times \log n$ sorting programs.