Side Talk: Memory Management in Merge Sort

Yufei Tao

ITEE
University of Queensland
In the class, we have learned that the merge sort algorithm sorts $n$ elements in $O(n \log n)$ time. In this side talk, we will discuss how to implement the algorithm using only $O(n)$ memory cells. In fact, as we will see, only two arrays—each of length $n$—are needed at any moment.
Recall:

**Merge Sort**

**Inductive Case.** The algorithm runs in three steps:

1. Recursively sort the first half of the array $S$.
2. Recursively sort the second half of the array.
3. Merge the two halves of the array into the final sorted sequence.
Example

Input:

38 28 88 17 26 41 72 83 69 47 12 68 5 52 35 9

First step, sort the first half of the array by recursion.

17 26 28 38 41 72 83 88 69 47 12 68 5 52 35 9

Second step, sort the second half of the array by recursion:

17 26 28 38 41 72 83 88 5 9 12 35 47 52 68 69

Third step, merge the two halves.

5 9 12 17 26 28 35 38 41 47 52 68 69 72 83 88
We can implement the merge step (i.e., Step 3 of Slide 3) as follows:

- Let $A$ be the input array
- Let $A_1$ be the first half (already sorted) of $A$
- Let $A_2$ be the second half (already sorted) of $A$
- Create another array $A'$ of length $n$
- Use $A'$ to perform the merging of $A_1$ and $A_2$
- Copy $A'$ to $A$
- Delete $A'$ (i.e., freeing up memory)
Example

Create array $A'$:

Appending 5, 9, 12, 17 to $A$, and so on:
Example

At the end of merging:

Copy $A'$ to $A$, and destroy $A'$:
Next, let us combine the above manual memory management with the automatic memory management done by the operating system to have a global view of how the memory consumption of merge sort changes. For this purpose, we will demonstrate the entire history of memory allocation for sorting 7 elements.
active portion of current recursion

local variables of your program

O(1) cells

two copies of your local variables
backtrack from recursion to the previous level

backtrack and then merging

backtrack and then merging
Side Talk: Memory Management in Merge Sort
Side Talk: Memory Management in Merge Sort
Side Talk: Memory Management in Merge Sort