Computação em Larga Escala

General Problems – Algorithmic analysis 4

António Rui Borges

Summary

- *Sorting a sequence of integers*
  - *Validation*
  - *Top-down and bottom-up approaches to solution*

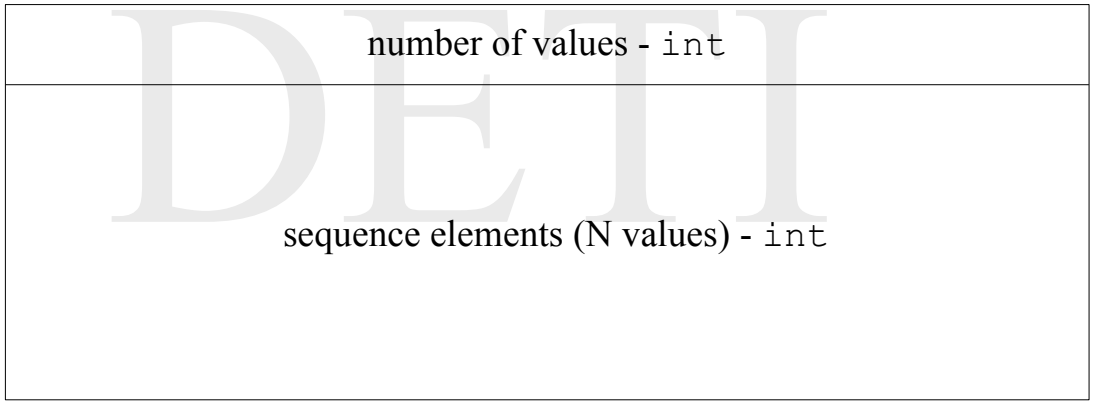
Sorting a sequence of integers - 1

Program validation

- `datSeq32.bin`
- `datSeq256K.bin`
- `datSeq1M.bin`
- `datSeq16M.bin`

Sorting a sequence of integers - 2

Format of binary files

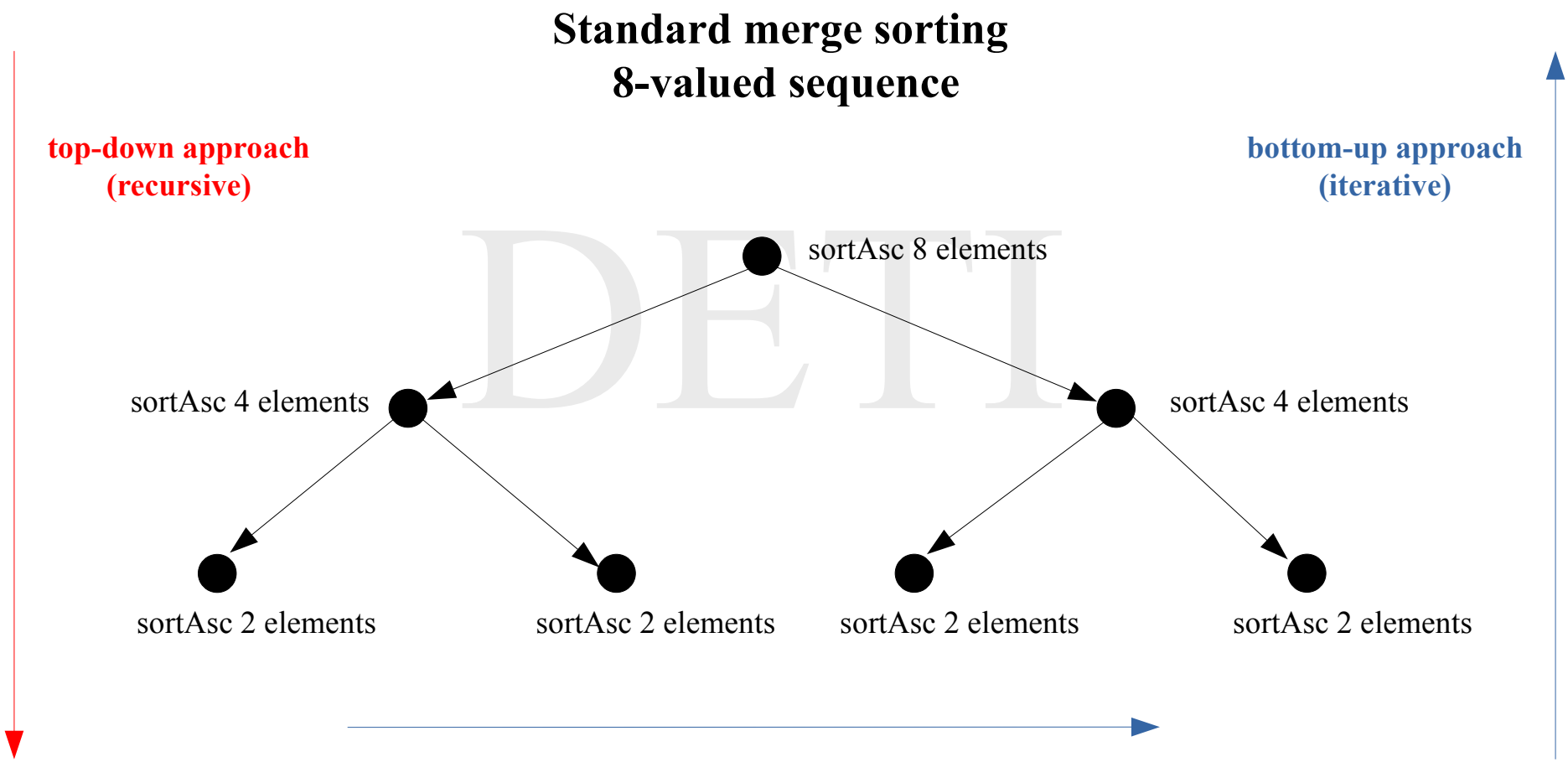


Sorting a sequence of integers - 3

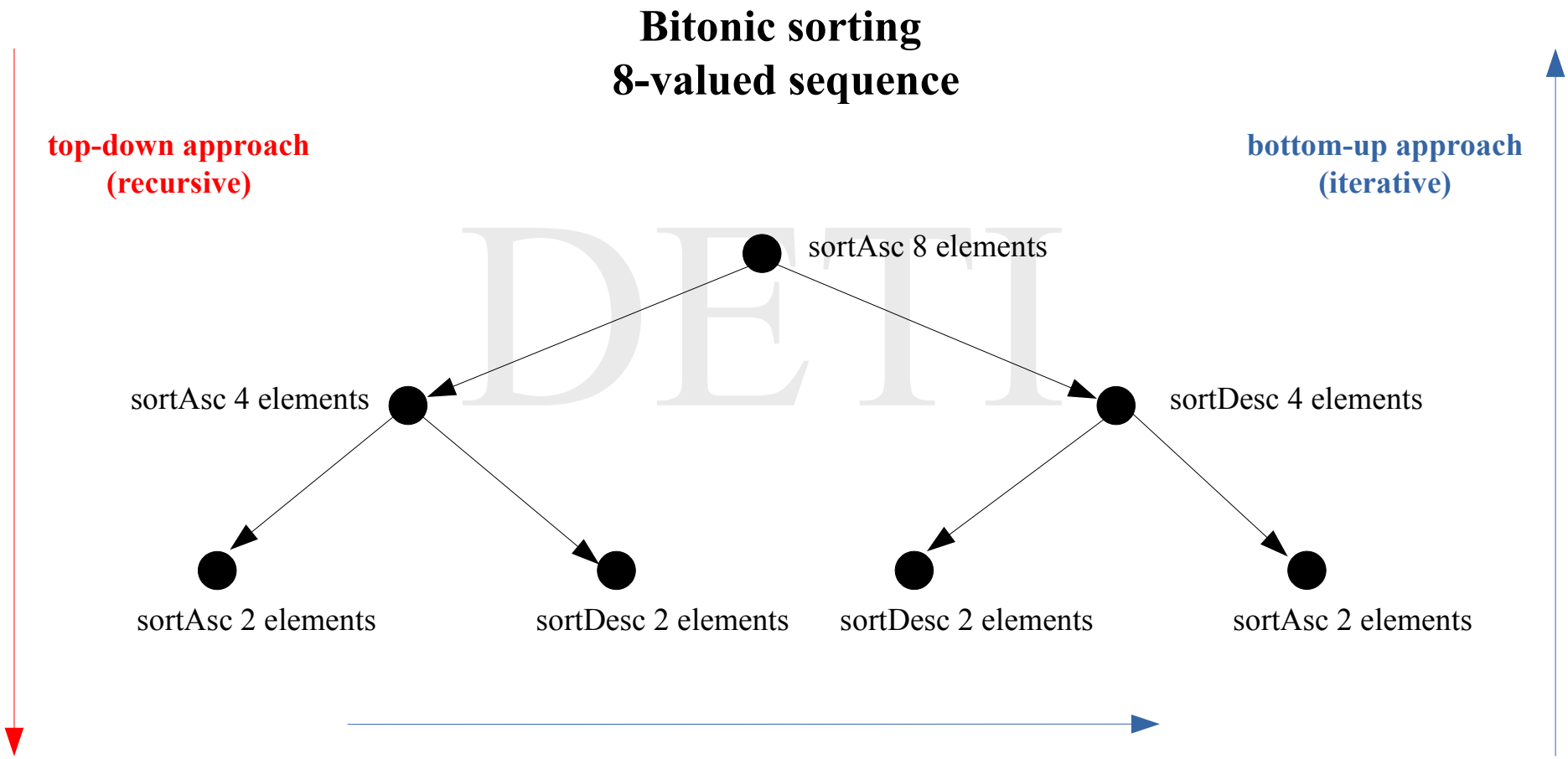
Code to be included for validation

```
for (i = 0; i < N - 1; i++)
  if (val[i] < val[i+1])
     { printf ("Error in position %d between element %d and %d\n",
               i, val[i], val[i+1]);
       break;
     }
  if (i == (N - 1))
     printf ("Everything is OK!\n");
```

Sorting a sequence of integers - 4



Sorting a sequence of integers - 5



Sorting a sequence of integers - 6

Total complexity

Standard merge sorting

$$\frac{N^2 + N\left(\log_2 N - 1\right)}{4} \text{ CAPS}$$

Bitonic sorting

$$\frac{N \log_2 N \log_2 2N}{4} \text{ CAPS}$$

with N equal to a power of 2.

Sorting a sequence of integers - 7

Final comments

Bitonic sorting, in the form it is usually presented, requires sorting to take place both in ascending and descending order. However, if one looks closer, this feature can be avoided and only sorting in ascending order is really required. Can you figure out what has to be done so that this becomes true?

Please, bear in mind that while either approach, recursive or iterative, is possible for a single-threaded solution, one always favors the iterative approach for the multithreaded solution.