NPTEL MOOC

PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON

Week 4, Lecture 2

Madhavan Mukund, Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Merge sorted lists

- Given two sorted lists `A` and `B`, combine into a sorted list `C`
  - Compare first element of `A` and `B`
  - Move it into `C`
  - Repeat until all elements in `A` and `B` are over
- Merging `A` and `B`

Analysis of Merge

How much time does Merge take?

- Merge `A` of size `m`, `B` of size `n` into `C`
- In each iteration, we add one element to `C`
  - Size of `C` is m+n
  - $m+n \leq 2 \max(m,n)$
- Hence $O(\max(m,n)) = O(n)$ if $m \approx n$

Merge Sort

To sort `A[0:n]` into `B[0:n]`

- If n is 1, nothing to be done
- Otherwise
  - Sort `A[0:n//2]` into `L` (left)
  - Sort `A[n//2:n]` into `R` (right)
  - Merge `L` and `R` into `B`

Analysis of Merge Sort …

- T(n): time taken by Merge Sort on input of size n
  - Assume, for simplicity, that $n = 2^k$
- $T(n) = 2T(n/2) + n$
  - Two subproblems of size n/2
  - Merging solutions requires time O(n/2+n/2) = O(n)
- Solve the recurrence by unwinding

Analysis of Merge Sort …

- $T(1) = 1$
- $T(n) = 2T(n/2) + n$

  $= 2 [2T(n/4) + n/2] + n = 2^2 T(n/2^2) + 2n$

  $= 2^2 [2T(n/2^3) + n/2^2] + 2n = 2^3 T(n/2^3) + 3n$

  …

  $= 2^j T(n/2^j) + jn$

- When $j = \log n$, $n/2^j = 1$, so $T(n/2^j) = 1$
  - $\log n$ means $\log_2 n$ unless otherwise specified!
- $T(n) = 2^j T(n/2^j) + jn = 2^{\log n} + (\log n)\, n = n + n \log n = O(n \log n)$

Variations on merge

- Union of two sorted lists (discard duplicates)
  - While `A[i] == B[j]`, increment `j`
  - Append A[i] to C and increment `i`
- Intersection of two sorted lists
  - If `A[i] < B[j]`, increment `i`
  - If `B[j] < A[i]`, increment `j`
  - If `A[i] == B[j]`
    - While `A[i] == B[j]`, increment `j`
    - Append A[i] to C and increment `i`
- Exercise: List difference: elements in A but not in `B`

Merge Sort: Shortcomings

- Merging A and B creates a new array C
  - No obvious way to efficiently merge in place
- Extra storage can be costly
- Inherently recursive
  - Recursive call and return are expensive