CS 170, Fall 2022

Contents

1. Algorithms with numbers

2. Divide-and-conquer Algorithms

2.a. Multiplication

Definition 2.1 (Integer Multiplication)

A divide-and-conquer algorithm for integer multiplication is defined as follows:

```
function mul(x(0b[1...k]), y(0b[1...h]))
%Input: Positive integers x, y in binary
%Output: x times y

n = max(size of x, size of y)
if n == 1: return x × y

x_L, x_R = x(0b[1...⌈n/2⌉]), x(0b[⌊n/2⌋...n])
y_L, y_R = y(0b[1...⌈n/2⌉]), y(0b[⌊n/2⌋...n])

P_1 = multiply(x_L, y_L)
P_2 = multiply(x_R, y_R)
P_3 = multiply(x_L + x_R, y_L + y_R)
return P_1 × 2^n + (P_3 − P_1 − P_2) × 2^(n/2) + P_2
```

Where $0b[1...k]$ denotes the binary string representing a number

2.b. Recurrence Relations

Theorem 2.2 (Master Algorithm)

If $T(n) = aT(n/b) + cn^k$ and $T(1) = c$ for some constants a, b, c and k, then

$$T(n) \in \begin{cases} O(n^k) & if\, a < b^k \\ O(n^k \log n) & if\, a = b^k \\ O(n^{\log_b a}) & if\, a > b^k \end{cases}$$

2.c. Mergesort

Definition 2.3 (Mergesort)

The Mergesort algorithm is defined as follows:

```
function mergesort(a[1...n])
  %Input: An array of numbers a[1...n]
  %Output: Sorted array a

  if n>1:
    return merge(mergesort(a[1...⌊n/2⌋]), mergesort(a[⌊n/2⌋+1...n]))
  else:
    return a

function merge(x[1...k], y[1...h])
  %Input: Two arrays of numbers (x[1...k], y[1...h])
  %Output: An array of numbers in x and y in ascending order

  if k=0: return y
  if l=0: return x
  if x[1] <= y[1]:
    return x[1] ∘ merge(x[2...k], y[1...h])
  else:
    return y[1] ∘ merge(x[1...k], y[2...h])
```

Where $\circ$ denotes concatenation.

The *merge* function above does a constant amount of work (concatenating two arrays) per recursive call, for a total running time of $O(k+l)$. Thus the calls to *merge* in *mergesort* are linear, we conclude that the overall time taken by *mergesort* is

$$T(n) = 2T(n/2) + O(n)$$

Recall the Master Algorithm, here $a = b^k$, and therefore

$$T(n) \in O(n \log n)$$

2.d. Medians

2.e. Matrix Mltiplication

Definition 2.4

2.f. Fast Fourier Transform