CS170–Spring 2020 — Homework 2 Solutions

Blackteaxx

2. Asymptotic Properties

(a) For all $c > 0$, we have following inequality:

$$\sum_{i=\frac{n}{2}}^{n} i^c \leq \sum_{i=1}^{n} i^c \leq n \cdot n^c$$

$$\frac{n}{2}^{c+1} \leq \sum_{i=1}^{n} i^c \leq n^{c+1}$$

We can throw the constant factor, and get the result $\sum_{i=1}^{n} i^c = \Theta(n^{c+1})$

(b)

$$\sum_{i=0}^{n} \alpha^i = \frac{1-\alpha^n}{1-\alpha}$$

According to the general result, we can get the asymptotic properties easily.

3. Median of Medians

(a) The worst case happens when the median comes in the first element of the array and the pivot chosen at each level of recursion is the last element.

The runtime is $\frac{n(n-1)}{2}$, a possible sequence pivot choice is $n, n-1, \cdots, 2$

(b) p is median of medians, so we have the following relations:

In step 3, the number of elements which are less than p, is $\frac{n}{2}$, and with the property of **Transitivity**, the number of elements which are less than p in Step 2, is at least $\frac{3n}{10}$

The opposite is similar.

(c)

$$T(n) \leq T(\frac{7}{10}n) + n$$

We can infer that:

$$T(n) \leq \mathcal{O}(n) + \frac{n}{5}\mathcal{O}(1) + T(\frac{n}{5}) + T(\frac{7}{10}n) \leq T(\frac{n}{5}) + T(\frac{7}{10}n) + \mathcal{O}(n)$$

assume that $T(n) \leq B \cdot n$

Basic Case: $T(1) = 1 \leq B$

Inductive Step: $T(n) \leq B \cdot n$, then we have $B \cdot \frac{7}{10}n + B \cdot \frac{n}{5} + cn \leq B \cdot n \Rightarrow \frac{B}{10} \geq c$

4. Werewolves

5. Hadamard matrices

(a)

$$H_0 = \begin{bmatrix} 1 \end{bmatrix}$$

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

(b)

$$H_2 \cdot v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4 \end{bmatrix}$$

(c)

$$u_1 = H_1(v_1 + v_2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$u_2 = H_1(v_1 - v_2) = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The result is equal to $H_2 \cdot v$

(d)

$$H_k \cdot v = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} H_{k-1}(v_1 + v_2) \\ H_{k-1}(v_1 - v_2) \end{bmatrix}$$

(e) We can produce a recursive algorithm

Algorithm 1 Hadamard matrix mutiplication

Require: $k \geq 0, v$
Ensure: $H_k v$
  if $k = 0$ **then**
    return v
  end if
  $v_1, v_2 =$ split v
  $u_1 =$ Hadamard matrix mutiplication$(k-1, v_1)$
  $u_2 =$ Hadamard matrix mutiplication$(k-1, v_2)$
  $u =$ combine $u1, u2$
  return u

Runtime Analysis:

$$T(n) = 2T(\frac{n}{2}) + \mathcal{O}(n) = \mathcal{O}(n \log n)$$

6. Warmup: FFT

(a) k is odd

(b) According the geometric meaning of the solution, we can build a mapping.

(c)

$$\omega_1 = e^{\frac{\pi k i}{n}}$$

$$\omega_2 = e^{\frac{\pi (n+k) i}{n}}$$