CS 301 - Homework 2

Ulaş Eraslan
25058

1- a) Modified version of Radix Sort is the following:

Radix Sort (A) {

- Find max. length in given list of word. // 6 (Furkan)
- Put "0" at the end of the words which their length is smaller than the max. length. // Reha00, Berk00, Husnu0

for (i = A.length(); i >= 1 ; i--) {

Counting Sort (A)

}

- Replace changed word with orginal ones. // Reha00 → Reha

}

b) List = HUSNU, BERK, FURKAN, REHA

1. Iteration

| HUSNU0 BERK00 REHA00 | FURKAN | } HUSNU0 BERK00 REHA00 FURKAN

2. Iteration

| BERK00 REHA00 | FURKAN | HUSNU0 | } BERK00 REHA00 FURKAN HUSNU0

3. Iteration

| REHA00 | BERK00 FURKAN | HUSNU0 | } REHA00 BERK00 FURKAN HUSNU0

4. Iteration

| REHA00 | BERK00 FURKAN | HUSNU0 | } REHA00 Berk00 FURKAN HUSNU0

5. Iteration

REHAOO
BERKOO

FURKAN
HUSNUO

} REHAOO BERKOO FURKAN HUSNUO

6. Iteration

BERKOO FURKAN HUSNUO REHAOO } BERKOO FURKAN HUSNUO REHAOO

⇓ After replacing with orginal words

BERK FURKAN HUSNU REHA

c) Finding Max. Length => $O(n)$

Putting 0 at the end of the words that are smaller than max => $O(n)$

Counting Sort with each iteration => $O(d(n+k))$ where d = max. length (6)

$k = 26$

Replacing with orginal ones => $O(n)$

$T(n) = O(n) + O(n) + O(d(n+k)) + O(n)$ where d and k are constant numbers

$T(n) = O(n)$ which is a linear.

2- a) WCL - Select consists of the following operations:

Ulaş Eraslan
25058

- dividing into groups => This takes constant time, $O(1)$
- finding medians => We need $\underbrace{\text{divide array to } k}$ part, sort and find median

$$O\left(\frac{n}{k}\right) * O(1) = O\left(\frac{n}{k}\right) = O(n)$$

- median of medians => We need to check each sub-group's median recursively by using WCL-Select so it takes $O\left(\frac{n}{k}\right)$

- partitioning => It takes $O(n)$ time

- Recursive call => In the worst case we go over $\dfrac{\left(2k - \left(\frac{k+1}{2}\right)\right) \cdot n}{2k}$

where $2k$ is the number of items which are on the left and right sides of the median.

$\frac{k+1}{2}$ is the number of elements which are guaranteed to be smaller than equal to median.

$$T(n) \leqslant O(1) + O\left(\frac{n}{k}\right) + T\left(\frac{n}{k}\right) + O(n) + T\left(\frac{(3k-1) \cdot n}{4k}\right).$$

for $k=5$ => $T(n) \leqslant O(1) + O\left(\frac{n}{5}\right) + T\left(\frac{n}{5}\right) + O(n) + T\left(\frac{14n}{20}\right)$

$$\leqslant T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + O(n)$$

b) for $k=7 \Rightarrow T(n) \leq T(\frac{n}{7}) + T(\frac{5n}{7}) + O(n)$

To show this is still liner time complexity, let's use substitution method.

Guess : $T(n) = O(n)$

Induction Hypothesis : $T(s) \leq c \cdot s$ for all $s < n$

So we need to show $T(n) \leq c \cdot n$

$$T(n) \leq c \cdot \frac{n}{7} + c \cdot \frac{5n}{7} + O(n)$$

$$\leq \frac{6nc}{7} + O(n)$$

$$\leq cn - \underbrace{\left(\frac{cn}{7} - O(n)\right)}_{>0,\ c>7,\ n \geq 1}$$

So $T(n) \leq c \cdot n$ ✓

c) To find the largest odd value that makes algorithm super-linear we may use substitution method.

Guess: $T(n) \leqslant c \cdot n$

Induction Hypothesis: $T(s) \leqslant c \cdot s$ where $s < n$

So let's show $T(n) \leqslant c \cdot n$

$$T(n) \leqslant \frac{c \cdot n}{k} + \frac{c \cdot (3k-1) \cdot n}{4k} + O(n)$$

$$\leqslant \frac{4cn}{4k} + \frac{(3k-1) \cdot cn}{4k} + O(n)$$

$$\leqslant \frac{(3k+3) \cdot cn}{4k} + O(n)$$

$$\leqslant c \cdot n - \left(\underbrace{\frac{(k-3) \cdot cn}{4k}}_{\text{to make this non-negative}} - O(n) \right)$$

$$\frac{k-3}{4k} > 0 \quad \text{so} \quad k > 3.$$

Therefore, largest odd value that makes algorithm super linear is $k = 3$.