

Data Structures and Algorithms

Exercise 4

Markus Innerebner, Romans Kasperovics
dsa(a)inf.unibz.it

Free University of Bozen - Bolzano, Italy

Thursday, March 19th, 2009

Assignment 04

Task 1 - Recurrences

Solve the following recurrences

- a) $T(n) = 15T(n/4) + n^2$
- b) $T(n) = bT(n-1) - (b-1)$
- c) $T(n) = 6T(n/2) + n^2$
- d) $T(n) = T(n-1) + \lg n$
- e) $T(n) = 9^c T(n/3) + n^{2c}$

Naming Conventions

- For each assignment create a folder *nn_yourlogin*
 - *nn* – is a number of the assignment (01, 02, ..., 10)
 - *yourlogin* – is your login to the university network
- Name your files as *task#.c* or *task#.pdf*, where '#' is the number of the corresponding task in the assignment, for example:

```
/ 01_minnerebner
   task12.pdf
   task3.c
   task4.c
```

- Put your name and student ID in all files
- Send the solution folder via e-mail archived as *nn_yourlogin.zip* or *nn_yourlogin.tar.gz* file, for example:

```
01_minnerebner.tar.gz
```

Assignment 04

Task 2 - Sorting

Use the QUICKSORT algorithm described in the lecture notes to sort the character array $A = \{ 'Q', 'U', 'I', 'C', 'K', 'S', 'O', 'R', 'T' \}$ in increasing order.

- a) Write a C program that uses the Hoare partitioning. Modify in the code PARTITION from slides 31/Week4 by setting the pivot element (*x*) to the most left side.
- b) Show the call stack and the state of the array after each call of PARTITION(*A*,*l*,*r*) for each partitioning method.

	Q	U	I	C	K	S	O	R	T
1									
2									
3									
4									
5									
6									
7									
8									

Assignment 04

Task 3 - Sorting

- Analyze the runtime of your algorithm.
- Illustrate worst and best case and give for each case the suitable example.
- How would you modify the algorithm to sort into non-increasing order?

Hints

Solving Recurrences

- Solve the following recurrence using:
 - Master method,
 - substitution method

$$T(n) = \begin{cases} 4 & , \text{if } n = 1 \\ 2T(n/2) + 5n + 1 & , \text{if } n > 1 \end{cases}$$

Solution Using Master Method

$$n^{\log_b a} = n \implies \text{Case 2}$$

$$f(n) = 5n + 1 \in \Theta(n) \implies T(n) \in \Theta(n \lg n)$$

Hints

Solving Recurrences

Solution Using Substitution Method

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + 5n + 1 = 2\left(2T\left(\frac{n}{4}\right) + 5\frac{n}{2} + 1\right) + 5n + 1 = \\ &4T\left(\frac{n}{4}\right) + 5n + 4 + 5n + 2 + 5n + 1 = \\ &4\left(2T\left(\frac{n}{8}\right) + 5\frac{n}{4} + 1\right) + 5n + 2 + 5n + 1 = 8T\left(\frac{n}{8}\right) + 5n + 4 + 5n + 2 + 5n + 1 \end{aligned}$$

$$2^i T\left(\frac{n}{2^i}\right) + i \cdot 5n + 2^i - 1$$

$$\frac{n}{2^i} = 1 \text{ when } i = \lg n$$

$$2^{\lg n} \cdot 4 + \lg n \cdot 5n + 2^{\lg n} - 1 = 5n \cdot \lg n + 6n - 1$$

Our Guess

$$T(n) \leq cn \lg n$$

Proof with Induction

- $T\left(\frac{n}{2}\right) \leq c \frac{n}{2} \lg \frac{n}{2}$

- ? $2 \cdot c \frac{n}{2} \lg \frac{n}{2} + 5n + 1 \leq cn \lg n$

Hints

Task 2: Quicksort. TODO add example