Parallel Algorithms Spring 2022

Problem Set 1

Prof. Nodari Sitchinava *Due: Wednesday, January 19, 2022 at 10:30am*

You may discuss the problems with your classmates, however **you must write up the solutions on your own** and **list the names** of every person with whom you discussed each problem.

Start **every** problem on a separate page. ***Any problem submitted by 11:59pm Friday January 14, 2022 will receive an additional 10% of the score you receive on that problem.***

1 Recursive Algorithm Analysis (10 pts)

Write down the recurrence, which defines the running time of the following algorithm and solve it using any method you like. Justify your answers.

```
function SillyRecursion(n)
    if n < 2
        return 1
    else
        x = 0
        for i = 1 to 4
            x = x + SillyRecursion(n/2)
        for i = 1 to n
            for j = 1 to n
                x = x + 1
        return x
```

2 Recurrences (15 pts)

Solve the following recurrences using any method you like. Show your work.

(a) **(5 pts)**
$T(n) = T(n/2) + 5$

(b) **(5 pts)**
$T(n) = T(n/2) + \sqrt{n}$

(c) **(5 pts)**
$T(n) = T(n/2) + n$

3 Proof By Induction (20 pts)

Consider the following recurrence:

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

Using induction prove that the above recurrence is bounded by $T(n) \leq 1 + \log^2 n$ for any positive integer n.[1]

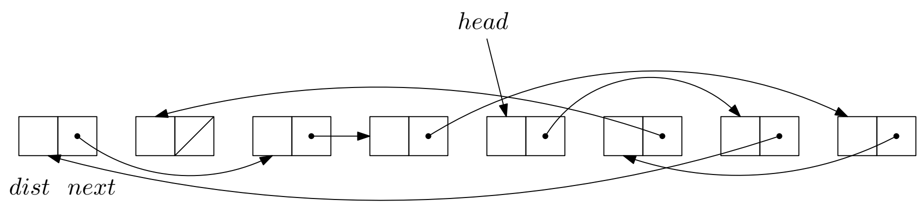
[1] $\log^2 n$ means $(\log n)^2$.

4 List Ranking (40 pts)

Let NODE be an abstract data type with the following specifications:

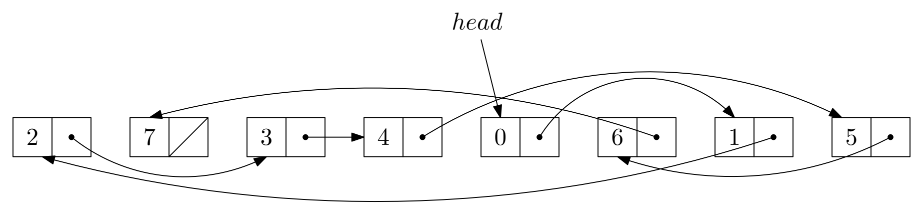
NODE:
  `int` *dist* ▷ distance value stored at each node
  NODE *next* ▷ pointer to the next node in the list

The node data structure can be used to compose a singly linked lists, for example:



(a) **(20 pts)** Design a (sequential) iterative (i.e., non-recursive) algorithm RANK(NODE v), which takes the head v of a linked list as input and updates the *dist* entry of every node of the linked list with the distance from the input node.

Thus, after the call to RANK(*head*) on the above linked list, it should look as follows:

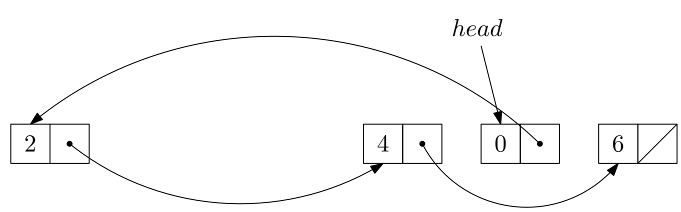


Use loop invariant to prove your algorithm's correctness. *Loop invariants are described in CLRS Chapter 2, in case you need to review them.*

(b) **(20 pts)** *Splicing* a node v out of a linked list involves removing v and connecting v's predecessor to v's successor.

Design a (sequential) algorithm SPLICE(NODE v), which takes the head of a linked list (a NODE object) as input and splices out every node whose rank is *odd*.

Thus, after the call to SPLICE(*head*) on the above linked list, it should look as follows:



Your algoirthm should not create any new nodes or linked lists, i.e., you should modify the existing linked list *in-place* by deleting existing nodes. You may assume that RANK(*head*) from part (a) had already been called on the head of the list before calling SPLICE(*head*)

5 Summations (15 pts)

Consider the following infinite series:

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \dots$$

(a) **(5 pts)** What is the next term in the series?

(b) **(5 pts)** Use the mathematical symbol Σ to write down the above summation. I.e., fill in all the necessary parts on the right hand side of the following equation:

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \dots = \Sigma$$

(c) **(5 pts)** What do the above series add up to?