CSE202 - DESIGN AND ANALYSIS OF ALGORITHMS

Week 9 - Amortization & Balance

November 27, 2022

Yubo Cai

ÉCOLE POLYTECHNIQUE

IP PARIS

EXERCISE 1

From the lecture we have that $T(m,n,r) = T(F,C) \leq T(F_+,C_+) + T(F_-,C_-) + m_+ + n$ and we have $T(F,C) \leq m + 2n\log^* r$, where $\log^* n$ is defined as the number of iterations of $\log_2$ before reaching ≤ 1.

From page 50 slides we can have the bound for the high forest gives

$$T(F_+,C_+) \leq m_+ + \frac{n}{2^{s-1}}\log^* r$$

Therefore we can have the bound condition

$$T(F,C) \leq T\left(F_-,C_-\right) + 2m_+ + n + 2\frac{n}{2^s}\log^* r$$

Also, we have the condition for m that $m_+ := m - m_-$

$$T(F,C) \leq T\left(F_-,C_-\right) + 2m - 2m_- + n + 2\frac{n}{2^s}\log^* r$$

$$T(F,C) - 2m \leq T\left(F_-,C_-\right) - 2m_- + n + 2\frac{n}{2^s}\log^* r$$

We can choose $s = \lceil \log_2 \log^* r \rceil$ in order to have $2\frac{n}{2^s}\log^* r = 2\frac{2}{\lceil \log^* r \rceil}\log^* r \leq 2n$, therefore we got

$$T(F,C) - 2m \leq T\left(F_-,C_-\right) - 2m_- + 3n$$

where by the definition of F_- we got the rank of the nodes in this forest at most $\leq s = \lceil \log_2 \log^* r \rceil$. By interating $T\left(F_-,C_-\right)$ on $log^* log^* r$ times we got

$$\begin{aligned} T(F,C) &\leq 2m + 2nlog^* log^* r - m_- + 3n \\ &= O(m \log^* \log^* n) \end{aligned}$$

In order to make $\log^* log^* n = 3$, which we got $\log^* n = 16$. Then we need to compute the function $f : x \to 2^x$ from $x = 1$ for 16 times. We assume this number is k which is really huge for computation. We can use a **Python** function to compute it.

log_Interation.py

```python
import math

def _log(x, base):
    return (int)(math.log(x) / math.log(base))

def recursiveLogStar(n, b):
    if (n > 1.0):
        return 1.0 + recursiveLogStar(_log(n, b), b)
    else:
        return 0

print(recursiveLogStar(1000000**1000000, 2)) # The output is 5.0
```

Since from the computation of the **Python** program, for 1000000^{100000} this huge number we got 5.0, so in order to get 16 the number n is tremendous. So we just assume k is the number such that $\log^* k = 16$ and make $n = k + 1$ in order to make the output larger than 3.