

Runtime analysis of the SLOWSORT algorithm

Lucio Santi

lsanti at dc.uba.ar

1 Introduction

This document provides a runtime analysis of the SLOWSORT algorithm[1]. Despite being presented in a satirical article, its recursion pattern turns out to be rather complex, which in turn makes it harder to analyze. On the other hand, the authors do not provide a thorough derivation of the claimed running time in the aforementioned article. The asymptotic bounds explored here do not match tightly with those, as they differ roughly by a linear factor.

2 Slowsort algorithm

The following figure shows the pseudocode of SLOWSORT :

```
SLOWSORT (A[1...n]):  
  If n = 1: return.  
  Let m :=  $\frac{n+1}{2}$ .  
  SLOWSORT (A[1...m]).  
  SLOWSORT (A[m+1...n]).  
  If A[n] < A[m]: swap A[n] and A[m].  
  SLOWSORT (A[1...n-1]).
```

3 Runtime analysis

3.1 Recurrence relation

Let $T(n)$ be the function that characterizes the worst-case running time of SLOWSORT when given an input array of size n . Being a recursive algorithm, we can define T recurrently like so:

$$T(n) = 2T(n/2) + T(n-1) + \Theta(1)$$

Note that we intentionally leave out base cases as we aim at finding asymptotic bounds for T . From this, we can see that $T(i) - T(i-1) = 2T(i/2) + \Theta(1)$ for a given i . Thus,

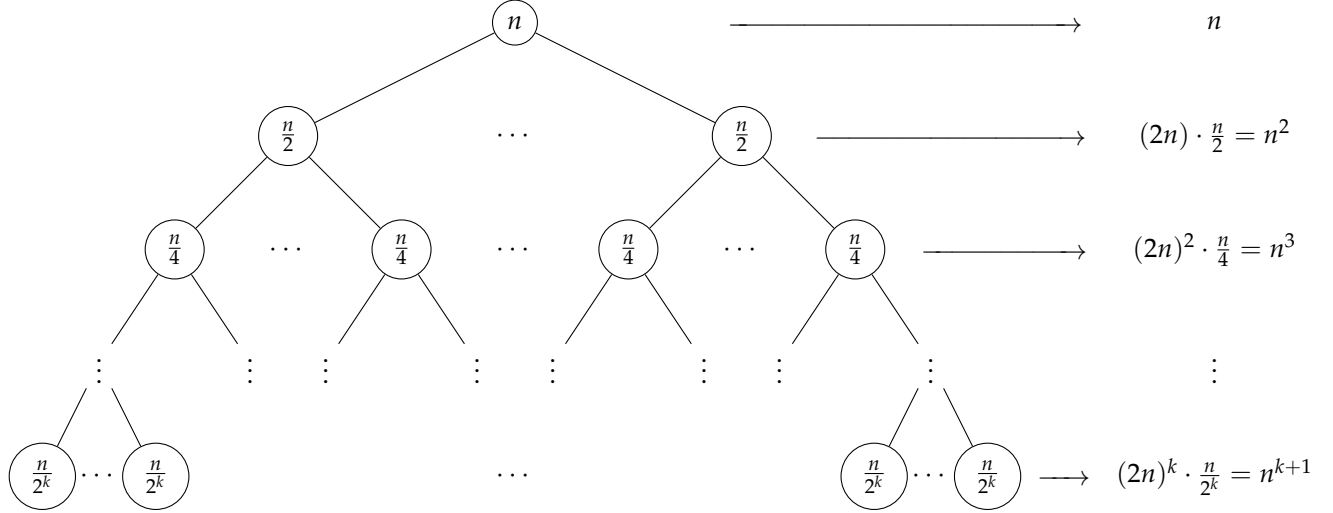
$$\begin{aligned} \sum_{i=1}^n T(i) - T(i-1) &= \sum_{i=1}^n 2T(i/2) + \Theta(1) \\ \Rightarrow T(n) - T(0) &= \sum_{i=1}^n 2T(i/2) + \sum_{i=1}^n \Theta(1) \\ \Rightarrow T(n) - T(0) &= \sum_{i=1}^n 2T(i/2) + \Theta(n) \\ \Rightarrow T(n) &= \sum_{i=1}^n 2T(i/2) + \Theta(n) \end{aligned}$$

3.2 Asymptotic bounds

We shall look now for asymptotic bounds for T . Let us consider first the following recurrence U :

$$\begin{aligned} U(n) &= \sum_{i=1}^n 2U(n/2) + \Theta(n) \\ &= 2n U(n/2) + \Theta(n) \end{aligned}$$

Clearly, we have that $T(n) \leq U(n)$ for all n . In order to find an asymptotic solution for U , let's have a look at its recursion tree:



From this, it follows that

$$\begin{aligned} U(n) &= \sum_{k=0}^{\lfloor \log_2(n) \rfloor} n^{k+1} \\ &= n \sum_{k=0}^{\lfloor \log_2(n) \rfloor} n^k \\ &= \frac{n}{n-1} (n^{\lfloor \log_2(n) \rfloor + 1} - 1) \\ &\in \Theta(n^{\log_2(n)+1}) \end{aligned}$$

This yields an asymptotic upper bound for T :

$$T(n) \in O(n^{\log_2(n)+1}) \quad (1)$$

In order to find an asymptotic lower bound, let's first rewrite T in the following convenient way:

$$T(n) = \sum_{i=0}^{n-1} 2T(n/2 - i/2) + \Theta(n)$$

Now, define the recurrence $L(n)$ as follows:

$$L(n) = \sum_{i=0}^{\lfloor \sqrt{n} \rfloor - 1} 2L(n/2 - i/2) + \Theta(n)$$

Such an L nearly satisfies every hypotheses required by the Akra-Bazzi theorem[2]. The only apparent exception is that the summation in the right-hand side goes up to a function of n , and the theorem states that this value should be a constant $k \in \mathbb{N}$. However, the proof outlined in [2] does not seem to rely

strongly on this fact¹. In what follows, then, we will use the Akra-Bazzi theorem to obtain an asymptotic solution for L .

Before proceeding, we observe that the index perturbation functions $h_i(n)$ are properly bounded:

$$\begin{aligned}
 |h_i(n)| &= \left| -\frac{i}{2} \right| \\
 &= \frac{i}{2} \\
 &\leq \frac{\lfloor \sqrt{n} \rfloor - 1}{2} \\
 &\leq \sqrt{n} \\
 &\in O\left(\frac{n}{\log^2 n}\right)
 \end{aligned} \tag{2}$$

(2) holds since $\frac{\sqrt{n}}{n/\log^2 n}$ goes to 0 as $n \rightarrow \infty$.

Now, we need to find the value of p to plug into the theorem and find a tight asymptotic bound for L :

$$\begin{aligned}
 \sum_{i=0}^{\lfloor \sqrt{n} \rfloor - 1} a_i b_i^p &= 1 \\
 \Leftrightarrow \sum_{i=0}^{\lfloor \sqrt{n} \rfloor - 1} 2 \frac{1}{2^p} &= 1 \\
 \Leftrightarrow \frac{\lfloor \sqrt{n} \rfloor}{2^{p-1}} &= 1 \\
 \Leftrightarrow \log_2(\lfloor \sqrt{n} \rfloor) &= p - 1 \\
 \Leftrightarrow p &= \log_2(\lfloor \sqrt{n} \rfloor) + 1
 \end{aligned}$$

With this in hand, we can calculate the following integral:

$$\begin{aligned}
 \int_1^n \frac{g(x)}{x^{p+1}} dx &= \int_1^n \frac{x}{x^{\log_2(\lfloor \sqrt{n} \rfloor) + 2}} dx \\
 &= \int_1^n \frac{dx}{x^{\log_2(\lfloor \sqrt{n} \rfloor) + 1}} \\
 &= \frac{-1}{\log_2(\lfloor \sqrt{n} \rfloor) x^{\log_2(\lfloor \sqrt{n} \rfloor)}} \Big|_1^n \\
 &= \frac{1}{\log_2(\lfloor \sqrt{n} \rfloor)} - \frac{1}{\log_2(\lfloor \sqrt{n} \rfloor) n^{\log_2(\lfloor \sqrt{n} \rfloor)}}
 \end{aligned}$$

Finally,

$$\begin{aligned}
 n^p \left(1 + \int_1^n \frac{g(x)}{x^{p+1}} dx \right) &= n^{\log_2(\lfloor \sqrt{n} \rfloor) + 1} \left(1 + \frac{1}{\log_2(\lfloor \sqrt{n} \rfloor)} - \frac{1}{\log_2(\lfloor \sqrt{n} \rfloor) n^{\log_2(\lfloor \sqrt{n} \rfloor)}} \right) \\
 &\in \Theta\left(n^{\log_2(\lfloor \sqrt{n} \rfloor) + 1}\right) \\
 &= \Theta\left(n^{\frac{\log_2(n)}{2} + 1}\right)
 \end{aligned}$$

Therefore,

¹If someone sufficiently qualified to confirm or disprove this assertion ever reads this, please let me know. Thanks in advance!

$$L(n) \in \Theta\left(n^{\frac{\log_2(n)}{2}+1}\right)$$

which, since $L(n) \leq T(n)$ for all n , implies that

$$T(n) \in \Omega\left(n^{\frac{\log_2(n)}{2}+1}\right) \tag{3}$$

Putting everything together, it follows from (1) and (3) that, for sufficiently large n , there exist some constants $c_1, c_2 > 0$ such that

$$c_1 n^{\frac{\log_2(n)}{2}+1} \leq T(n) \leq c_2 n^{\log_2(n)+1}$$

References

- [1] A. Broder and J. Stolfi, "Pessimal algorithms and simplicity analysis," *ACM SIGACT News*, vol. 16, pp. 49–53, 1984.
- [2] T. Leighton, "Notes on better master theorems for divide-and-conquer recurrences," in *Lecture notes*, MIT, 1996.