1 Analysis of Algorithms

Let $\mathcal{F}$ denote the set of all functions from $\mathbb{N}$ to $\mathbb{R}^+$.

For any $f \in \mathcal{F}$, let

$$O(f) = \{g \in \mathcal{F} \mid \exists c \in \mathbb{R}^+ . \exists b \in \mathbb{N} . \forall n \in \mathbb{N} . (n \geq b \text{ IMPLIES } g(n) \leq cf(n)\}.$$

For an algorithm A, let $t(I)$ be the number of steps the algorithm A takes to halt on input I.

And what is a step?

Pick 1 or 2 operations such that the total number of operations performed by A is the same as the number of these opoerations performed by A, to within a constant factor.

1.1 Properties of O Notation

- For any $c \in \mathbb{R}^+ \cup 0$, $cf(n) \in O(f(n))$ and $f(n) \in O(cf(n))$.
- If $\lim_{n\to\infty} \frac{h(n)}{g(n)} = 0$, then $g(n) + h(n) \in O(g(n))$
- If $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in (h(n))$
- If $f_1 \in O(g_1)$ and $f_2 \in O(g_2)$, then $f_1 + f_2 \in O(g_1 + g_2)$.
- $\max\{f, g\} \in O(f + g)$
- $f + g \in O(\max\{f, g\})$
- If $f_1 \in O(g_1)$ and $f_2 \in O(g_2)$, then $f_1 f_2 \in O(g_1 g_2)$
- Leh $a + b$ be constant. If $a < b$,then $n^a \in O(n^b$
- If $1 < a < b$, then $a^n \in O(b^n)$, but $b^n not \in O(a^n)$
- For all $a, b > 1$, $\log_a(n) \in O(\log_b(n))$.

Example 1.1

Consider the following algorithm $LS(L, x)$ such that, if x occurs in L, the algorithm returns an index of L at which x occurs. Otherwise, return 0. Let L be an array with the index of 1:

i <- 1

while $i \leq$ length(L) do

if $L[i] = x$

then return i

i < i+1

end while

return 0.

Now, count the number of comparisons with x. Suppose that each iteration of the loop performs $O(1)$ steps (assuming LENGTH takes $O(1)$, and outside the loop $O(1)$ steps are performed.

Now, we can express the complexity as a function of the input size:

$$T_A : \mathbb{N} \to \mathbb{N} \tag{1}$$
$$T_A(n) = \max\{t_A(I) \mid \text{size}(I) = n\}, \tag{2}$$

which gives the worst case time complexity of an algorithm A. For LS, size$((L, x)) =$ length(L)

Example 1.2

Now we can estimate the average cas time complexity. Define $T'_A : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$, where $T'_A(n) = \mathbb{E}[t_A]$, where the expectation is taken over a probability space of all inputs of size n. If all inputs of size n are equally likely, then $T'_A(n) = \frac{\sum\{t_A(I) \mid \text{size}(I)=n\}}{\#\{I \mid \text{size}(I)=n\}}$