

Introduction to Algorithms: a Workbook

Howard A. Blair

© Copyright 2009
All rights reserved.

Permission to download and print is granted.
No printed version of this document may be sold.

Topic I: Asymptotic Notation.

1. **Problem 1:** Corman, et. al.: Problems 3-2, 3-3, 3-4, pp. 58–59.

Topic II: Algorithm correctness.

1. **Exercises:** Corman, et. al.: Problems 2-2, 2-3, pp. 38–39.

Topic III: Computational Complexity Classes.

1. **Roadmap:**

Sequential computational time complexity classes depend on:

Measuring elapsed time during a computation.

The measure depends on:

A universal model of sequential computation.

One such model is the *Turing machine*.

Formal languages.

Decision Problems.

Therefore, we will examine these ideas, each in their turn.

2. **Turing machines:** Here is a formal definition of a Turing machine according to

John Hopcroft and Jeffrey Ullman, (1979). Introduction to Automata Theory, Languages and Computation (1st ed.). AddisonWesley, Reading Mass. ISBN 0-201-02988-X

Definition: A Turing machine M is a 7-tuple: $\langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ where

- Q is a finite set of states. Uh-huh. What's a state?
- Γ is a finite set of the tape alphabet/symbols. What's a tape, what's an alphabet, what's a symbol?
- $b \in \Gamma$ is called *blank*. Note: b is a formal parameter in this definition. We can designate any symbol we want in an application to be the blank. The blank is the only symbol allowed to occur on the tape infinitely often at any step during the computation. By the way, what's a computation?
- $\Sigma \subseteq (\Gamma - \{b\})$. Σ is called the set of input symbols.
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is a partial function called the transition function, where L is left shift, R is right shift.
- $q_0 \in Q$. q_0 is called the *initial* state.
- $F \subseteq Q$. F is called the set of *final* (or, synonymously, *accepting*) states.

3. **Exercise:** Here is the first of the 5-state Busy Beaver Turing machines at <http://ironphoenix.org/tril/tm>

```

1,- 2,1,>
1,1 3,-,<
2,- 3,1,>
2,1 4,1,>
3,- 1,1,<
3,1 2,-,>
4,- 5,-,>
4,1 H,1,>
5,- 3,1,<
5,1 1,1,>

```

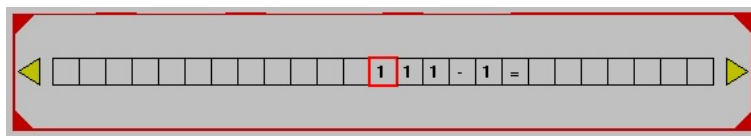
Identify the components of the 7-tuple for this example in Hopcroft and Ullman's definition.

4. **Alternative Definition:** A Turing machine M is a 7-tuple: $\langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ where

- Q is a finite set. The members of Q are called *states*.
- Γ is a finite set. Γ is called the *tape alphabet*. The members of Γ are called *symbols*.
- The *tape* of M is the set of integers \mathbb{Z} .
- $b \in \Gamma$ is called *blank*.
- $\Sigma \subseteq (\Gamma - \{b\})$. Σ is called the set of *input symbols*.
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is a partial function called the transition function, where
 - $L: \mathbb{Z} \rightarrow \mathbb{Z}$ by $L(n) = n - 1$. L is called left-shift.
 - $R: \mathbb{Z} \rightarrow \mathbb{Z}$ by $L(n) = n + 1$. R is called right-shift.
- $q_0 \in Q$. q_0 is called the *initial state*.
- $F \subseteq Q$. F is called the set of *final* (or, synonymously, *accepting*) states.

5. **Computations:** If we watch a Turing machine perform a computation, what we see at any given step is the tape, the contents of the tape, the location on the tape of the Turing machine's tape head and the internal state of the tape head. The the next step we see the same thing, but the tape contents, the tape head's internal state and the location of the tape head have changed slightly. The tape together with its contents can be represented by a function from the tape (i.e the integers) to the tape alphabet.

6. **Tape contents:** The tape and its contents corresponding to a particular Turing machine is pictured below. It is derived from the *Subtractor* Turing machine program available at <http://ironphoenix.org/tril/tm/>



Assume the leftmost 1 is in cell number 0. The representation of the tape contents by a function from the tape to the tape alphabet is then given by the function, call it γ , defined by the following input/out table:

$n < 0$	\flat
0	1
1	1
2	1
3	$-$
4	1
5	$=$
$n > 5$	\flat

where \flat is the symbol we are using to represent the blank symbol of the tape alphabet.

The above image also happens to show the location of the Turing machine's tape head. Even if we knew the Turing machine's program, we still do not know what the Turing machine will do with these tape contents, because we do not know, from the image, what the internal state of the tape head is. (This is a deficiency of the display available at the ironphoenix website.) Given a Turing machine, i.e. to be given δ and all of the other information that defines a Turing machine, then, to know what the Turing machine is going to do at a given computation step, we need to know

$$(\gamma, q, \text{loc})$$

where γ is the tape contents, q is the internal state of the tape head, and loc is the location of the tape head. We will call this triple a *configuration*. Some authors use the term *global state*. Some authors (Rogers '67; cf. course bibliography) represent the triple differently and call it an *instantaneous description*. This latter terminology refers to the fact that a configuration represents a computation at a given *instant*.

7. **Compact support:** There is more to say about γ , the tape contents. All but a finite part of the tape at any step in a computation is blank, since the blank is, by definition, the symbol that is allowed to occur infinitely often on the tape. The best way to characterize this restriction is with the concept of *compact support*. This concept transfers to continuous domains of application, while the more straightforward idea of simply constraining the blank symbol to be the only symbol allowed to occur infinitely often on the tape, does not.

Think of the set of integers \mathbb{Z} a metric space: i.e. a set with a notion

of distance defined between any two *points* (i.e. elements) in the set. The distance notion is called a *metric*. The standard metric on the integers is that the distance between two integers is the absolute value of their difference. For example, the distance between -1 and 4 is $|-1 - 4| = |-5| = 5$.

In our present application, a subset A of the set of integers is finite iff it is *bounded*.

The term *bounded* means that for any point p in \mathbb{Z} , there is a real number b such that the distance from p to any point in A is less than or equal to b . It also happens that a finite set of integers is *closed* in this metric space. (We won't get into details about that.) There is a still more general concept, called *compactness*, that transfers to continuous spaces and even continuous spaces without metrics. (We will skip the details.)

The bottom line is that a set of integers is finite if, and only if, it is compact in the set of integers considered as a metric space with the standard metric.

One last concept is needed for the concept of compact support: the concept of the support of a function: The *support* of a function $\gamma : \mathbb{Z} \rightarrow \Gamma$ is the set $\{n \mid \gamma(n) \neq b\}$.

Definition: $\gamma : \mathbb{Z} \rightarrow \Gamma$ has *compact support* iff the support of γ is compact.

It is the functions $\gamma : \mathbb{Z} \rightarrow \Gamma$ that have compact support that both (i) represent tape contents in which the blank symbol is the only symbol allowed to occur infinitely often, and (ii) involve concepts that transfer to continuous spaces such as multidimensional spaces over the real and complex numbers.

- 8. The transition from one configuration to the next:** The Turing machine's transition function δ determines a one-step transition *partial* function Δ that maps a configuration (γ, q, loc) to the "next" configuration $(\gamma', q', \text{loc}')$ if there is one.

9. **The type of a configuration:** Let $\mathbb{Z} \xrightarrow{c.s.} \Gamma$ be the set of functions from the integers \mathbb{Z} to the tape alphabet Γ that have compact support. (Recall that Q is the set of states of the Turing machine.) A configuration (γ, q, loc) is therefore a member of the set of triples

$$(\mathbb{Z} \xrightarrow{c.s.} \Gamma) \times Q \times \mathbb{Z}$$

I.e. this is the set of all configurations that are possible for the machine, although not necessarily reachable from an initial configuration during a computation.

10. **The type of Delta:**

$$\Delta: (\mathbb{Z} \xrightarrow{c.s.} \Gamma) \times Q \times \mathbb{Z} \rightarrow (\mathbb{Z} \xrightarrow{c.s.} \Gamma) \times Q \times \mathbb{Z}$$

The symbol \rightarrow indicates a partial function.

11. **Constructing the definition of Delta, the first step:** Given a configuration (γ, q, loc) , let the γ' be the tape contents of the configuration that results from one transition step of the Turing machine, i.e. the next tape contents. Let q' be the next state, and let loc' be the next location. The contents of the tape at location loc is

$$\gamma(\text{loc})$$

Also,

$$\delta(q, \gamma(\text{loc})) = (q', \gamma'(\text{loc}), \sigma)$$

where σ is a shift, i.e. $\sigma \in \{L, R\}$. The next location loc' is given by

$$\text{loc}' = \sigma(\text{loc})$$

Therefore,

$$\Delta(\gamma, q, \text{loc}) = (\gamma', q', \sigma(\text{loc}))$$

12. **Exercise:** Here is the configuration of the subtracter Turing machine given in item 6, assuming the state is 2:

$$(\gamma, 2, 0)$$

(See item 6 for γ .) Here is the subtracter Turing machine's program:

1,- 1,-,>
 1,1 1,1,>
 1,- 1,-,>
 1,= 2,-,<
 2,1 3,=,<
 2,- H,-,<
 3,1 3,1,<
 3,- 4,-,<
 4,- 4,-,<
 4,1 1,-,>

What is

γ' ?
 q' ?
 σ ?
 $\sigma(\text{loc})$?

13. **Partial function variants:** (We need this idea to understand how the next tape contents γ' is related to the current tape contents γ .) A *variant* of a partial function $f: A \rightarrow B$ is the partial function that results from changing the value returned by f for one particular input a . A notation that is sometimes used to describe a variant of f is given in the following definition:

$$(f \mid a \mapsto b)(x) = \begin{cases} f(x) & \text{if } x \neq a \\ b & \text{if } x = a \end{cases}$$

We allow for the possibility that $b = \perp$. That is, the variant of f might output no value corresponding to input a . We also allow f to be variant of itself. In that case, $b = f(a)$.

14. **Exercise:** Simplify the following:

$$((f \mid a \mapsto b) \mid a \mapsto c)$$

15. **Constructing the definition of Delta, the second step:** The next tape contents γ' is a variant of current tape contents at input loc . Specifically,

$$\gamma' = (\gamma \mid \text{loc} \mapsto s)$$

where $\delta(q, \gamma(\text{loc})) = (q', s, \sigma)$. Therefore,

$$\Delta(\gamma, q, \text{loc}) = ((\gamma \mid \text{loc} \mapsto s), q', \sigma(\text{loc}))$$

16. **Exercise:** Calculate $(\gamma \mid \text{loc} \mapsto s)$ and verify that it is equal to γ' .
17. **Projections:** [This idea will allow us to extract q' , s , and σ from $\delta(q, \gamma(\text{loc}))$]. Let A_1, \dots, A_n be nonempty sets, not necessarily distinct. In other words, sets may be repeated in this sequence. Consider

$$(a_1, \dots, a_n) \in A_1 \times \dots \times A_n$$

Suppose we want to extract the element a_j at the j^{th} position in (a_1, \dots, a_n) by applying a function to (a_1, \dots, a_n) . Such a function is called a *projection*. Here is a notation for the projection function that extracts the j^{th} member of an n -tuple (a_1, \dots, a_n) :

$$((a_1, \dots, a_n))_j$$

This notation for projection functions is used in various more advanced books on mathematical logic (but the starting index is 0 in those books, not 1). It is *overloaded*. That means the same notation is used regardless of the type of the n -tuple. Why does the notation use double parentheses? It really only seems to, but doesn't. Suppose v is an n -tuple from which we to extract the j^{th} member. Using the notation we would denote the j^{th} member of v by $(v)_j$. Suppose $v = (a_1, \dots, a_n)$. Then, $(v)_j = ((a_1, \dots, a_n))_j$.

18. **Constructing the definition of Delta, the last step:**

$$\begin{aligned} q' &= (\delta(q, \gamma(\text{loc})))_1 \\ s &= (\delta(q, \gamma(\text{loc})))_2 \\ \sigma &= (\delta(q, \gamma(\text{loc})))_3 \end{aligned}$$

Therefore,

$$\begin{aligned} &\Delta(\gamma, q, \text{loc}) \\ &= \\ &((\gamma \mid \text{loc} \mapsto (\delta(q, \gamma(\text{loc})))_2), (\delta(q, \gamma(\text{loc})))_1, (\delta(q, \gamma(\text{loc})))_3(\text{loc})) \end{aligned}$$

19. **Exercise:** Carefully verify that

$$((\gamma \mid \text{loc} \mapsto (\delta(q, \gamma(\text{loc})))_2), (\delta(q, \gamma(\text{loc})))_1, (\delta(q, \gamma(\text{loc})))_3(\text{loc}))$$

is equal to

$$(\gamma', q', -1)$$

Answers to Exercises, Questions and Problems

Solutions 1: Corman, et. al.: Problem 3-2, p. 58.

a) $\lg^k = o(n^\epsilon)$. Why? Instead of starting from a formula, consider: Let $\delta > 0$. Choose m such that $m \leq \frac{\epsilon}{k} 2^m + \lg \delta$. Then, for all n' such that $\lg n' \geq m$, $\lg n' \leq \frac{\epsilon}{k} 2^{\lg n'} + \lg \delta = \frac{\epsilon}{k} n' + \lg \delta$. Now choose n_0 such that $\lg \lg n_0 \geq m$. Then, for all $n \geq n_0$,

$$\lg \lg n \leq \frac{\epsilon}{k} \lg n + \lg \delta$$

Therefore,

$$\lg \lg^k n = k \lg \lg n \leq \epsilon \lg n + \lg \delta = \lg n^\epsilon + \lg \delta$$

$\lg x$ is a monotonic function and so is e^x . Therefore, $x \leq y$ iff $\lg x \leq \lg y$. Hence, for any n such that $\lg \lg n \geq m$, where $m \leq \frac{\epsilon}{k} 2^m + \lg \delta$,

$$\lg^k n \leq \delta n^\epsilon$$

Thus,

$$\lg^k n = o(n^\epsilon)$$

b) n^k vs. c^n . Given $\epsilon > 0$, for any m such that

$$m \geq \frac{\ln 2 - \ln \epsilon}{\ln(1 + \epsilon)}$$

we have

$$m \ln(1 + \epsilon) \geq \ln \frac{2}{\epsilon}$$

It follows that

$$(1 + \epsilon)^{m+1} - (1 + \epsilon)^m \geq 2$$

By induction,

$$(1 + \epsilon)^{m+j} - (1 + \epsilon)^m \geq 2j$$

For any j_0 such that

$$m - (1 + \epsilon)^m \leq j_0$$

we have

$$m + j_0 - (1 + \epsilon)^m \leq 2j_0 \leq (1 + \epsilon)^{m+j_0} - (1 + \epsilon)^m$$

and then

$$m + j_0 \leq (1 + \epsilon)^{m+j_0}$$

Therefore, for any $n \geq m + j_0$

$$n \leq (1 + \epsilon)^n$$

Now, let $\epsilon = c^{\frac{1}{2k}} - 1$. Then

$$n \leq (c^{\frac{1}{2k}})^n$$

Equivalently,

$$(n^k)^2 \leq c^n$$

for any $n \geq m + j_0$, where m and j_0 are chosen corresponding to $\epsilon = c^{\frac{1}{2k}} - 1$. Let a be any positive real number. For any sufficiently large n ,

$$an^k \leq (n^k)^2 \leq c^n$$

Therefore,

$$n^k = o(c^n)$$

The preceding argument depended on an argument by mathematical induction. We had established

$$(1 + \epsilon)^{m+1} - (1 + \epsilon)^m \geq 2$$

and we claimed

$$(1 + \epsilon)^{m+j} - (1 + \epsilon)^m \geq 2j$$

followed by induction. We include the induction proof here: We must prove:

for every every natural number $j > 0$ [Sentence(j)]

where Sentence(j) is

$$(1 + \epsilon)^{m+j} - (1 + \epsilon)^m \geq 2j$$

So, we must prove

Sentence(1), Sentence(2), \dots , Sentence(n), Sentence($n + 1$), \dots

i.e., we must prove

for all n , Sentence'(n)

where Sentence'(n) is Sentence($n + 1$). The principle of mathematical induction for a sentence $P(n)$ depending on a parameter n is

if (if $P(0)$ and for all n [if $P(n)$ then $P(n + 1)$]) then for all n , $P(n)$

The point is that if we succeed in proving

$$P(0)$$

and in proving

$$\text{for all } n[\text{if } P(n) \text{ then } P(n+1)]$$

then by the principle of mathematical induction,

$$\text{for all } n, P(n)$$

must follow.

For the present application, we'll take for $P(n)$ the sentence

$$(1 + \epsilon)^{m+(n+1)} - (1 + \epsilon)^m \geq 2(n+1)$$

So we must prove $P(0)$; i.e.

$$(1 + \epsilon)^{m+(0+1)} - (1 + \epsilon)^m \geq 2$$

and we must prove that for every n ,

$$\begin{aligned} \text{if } (1 + \epsilon)^{m+(n+1)} - (1 + \epsilon)^m &\geq 2(n+1), \\ \text{then } (1 + \epsilon)^{m+((n+1)+1)} - (1 + \epsilon)^m &\geq 2(n+2) \end{aligned}$$

We already proved $P(0)$. Actually - this important to the argument we are giving here - we proved $P(0)$ for any m greater than a certain threshold m_0 . To prove that if $P(n)$ then $P(n+1)$, we temporarily assume $P(n)$ i.e. we assume

$$(1 + \epsilon)^{m+(n+1)} - (1 + \epsilon)^m \geq 2(n+1)$$

(This is the *induction assumption*.) It then remains to prove $P(n+1)$; i.e. it remains to prove

$$(1 + \epsilon)^{m+((n+1)+1)} - (1 + \epsilon)^m \geq 2(n+2)$$

Because we proved

$$(1 + \epsilon)^{m+1} - (1 + \epsilon)^m \geq 2$$

for every m greater than m_0 , we have

$$(1 + \epsilon)^{m+((n+1)+1)} - (1 + \epsilon)^{m+(n+1)} \geq 2$$

Therefore,

$$\begin{aligned}
& (1 + \epsilon)^{m+((n+1)+1)} - (1 + \epsilon)^m \\
= & \\
& (1 + \epsilon)^{m+(n+1)} - (1 + \epsilon)^m + (1 + \epsilon)^{m+((n+1)+1)} - (1 + \epsilon)^{m+(n+1)} \\
\geq & \\
& 2(n + 1) + 2 \\
= & \\
& 2(n + 2)
\end{aligned}$$

which is what remained to be proved.

c) \sqrt{n} vs. $n^{\sin n}$. For infinitely many values of n , $\sin n$ is close to -1 , and for infinitely many values of n , $\sin n$ is close to, but is never greater than 1. Therefore, for infinitely many values of n , $n^{\sin n}$ is close to 0, and for infinitely many values of n , $n^{\sin n}$ is close to n , but is never greater than n . Formally,

$$\liminf_{n \rightarrow \infty} \frac{n^{\sin n}}{\sqrt{n}} = 0$$

and

$$\limsup_{n \rightarrow \infty} \frac{n^{\sin n}}{\sqrt{n}} = \infty$$

The first of these, the limit infimum, prevents the conclusion that

$$n^{\sin n} = \Omega(\sqrt{n})$$

The second, the limit supremum, prevents

$$n^{\sin n} = O(\sqrt{n})$$

Therefore, \sqrt{n} and $n^{\sin n}$ have incomparable growth rates.

d) 2^n vs. $2^{n/2}$.

$$\lim_{n \rightarrow \infty} \frac{2^n}{2^{n/2}} = \infty$$

Therefore,

$$2^n = \omega(2^{n/2})$$

e) $n^{\lg c}$ vs. $c^{\lg n}$.

$$\lg n^{\lg c} = \lg c \lg n = \lg n \lg c = \lg c^{\lg n}$$

Therefore, since \lg is a 1 to 1 function,

$$n^{\lg c} = c^{\lg n}$$

Therefore,

$$n^{\lg c} = \Theta(c^{\lg n})$$

f) $\lg n!$ vs. $\lg n^n$.

$$\begin{aligned}\lg n! &= \sum_{k=2}^n \lg k \\ &\geq \int_1^n \lg x \, dx \\ &= (x \lg x - x) \Big|_1^n \\ &= (n \lg n - n) - (1 \lg 1 - 1) \\ &= n \lg n - (n - 1) \\ &= n(\lg n - 1) + 1 \\ &= \Theta(n \lg n) \\ &= \Theta(\lg n^n)\end{aligned}$$

△