Fundamentals

RAM code for $\sum_{i=1}^{n} i$:

1. n = 2
2. R_sum = 0
3. R_count = 1
5. 1 > 2
6. not true
8. R_tmp = 0 (Load necessary!)
9. R_tmp = R_Tmp + R_count = 0 + 1
10. R_sum = 1

12 R_tmp2 = 1
13 R_tmp2 = 2
14 R_count = 2

15 go to line 4

5 2 > 2
6 not true

8 R_tmp = 1
9 R_tmp = R_Tmp + R_count = 1 + 2
10 R_sum = 3

12-14: R_count = 3
15 go to line 4
5 3 > 2
6 true → go to line 16
17 print 3

Theorem 1: Iteration-Sort correctly solves the sorting problem.

Proof: Observe that it is in fact an algorithm:

- "unambigously" described.
- finite input $A = A(0) .. A(n-1)$
- finite output —"—
- "memory" n numbers must be stored + 2 variables i, j = finite.
- **terminates**

FOR runs till $j = n$ stops

WHILE: $i \leq j$ (L3)

i decrement (L6) $\Rightarrow i < 0$ at some iteration ($< n$ iteration)

$\Rightarrow$ terminates

$\Rightarrow$ Alg. terminates.

---

proof now correctness:

We use "loop invariants" = statement that is true across multiple iterations of a loop.

$A = (x_0 \cdots x_{n-1})$

in the following we write $A(i..j)$ for the elements $A[i] .. A[j]$, $0 \leq i \leq j \leq n-1$

loop-invariant $P(j)$ = at each step, $A(0..j-1)$ "consist" of all original elements $x_0 .. x_{i-1}$ but in sorted order.

Show: (I) $P(j)$ holds at start of j-th iteration of FOR-loop

(II) $P(j)$ —" at end of j-th iteration of FOR loop

(III) $P(j)$ holds when alg. terminates.

$j = 1$ (= first iteration) : at start $A(0,0) = A[0]$ sorted, since it contains only 1 element, i.e., $P(1)$ is true.

at this point we can assume that $P(j)$ is true at start of j-th iteration of FOR-loop.
$\Rightarrow$ (I) holds

Show now that (II) holds.

2 cases : WHILE loop starts or does not run in jth iterat. of FOR loop

L3: $i = j - 1$

By assumption, (I) $P(j)$ holds:
(*) $A(0..j-1)$ is sorted & containing $x_0 .. x_{j-1}$
$A(0) \leq A(1) \leq .. \leq A(j-1)$

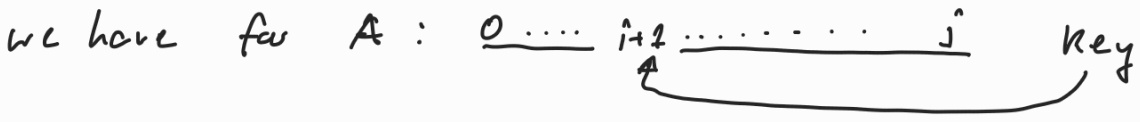
if WHILE-loop doesnot run
$\Rightarrow$ Since $i \geq 0$ as $j \geq 1$, we have $A(i) \leq key$. in L4.
$\Rightarrow$ L7 : $A(i+1) = key$, $i+1 = j \Rightarrow A(j) = key$
(*) $\Rightarrow$ $A(0) \leq A(1) \leq .. A(j-1) \leq A(j)$
$\Rightarrow$ $A(0..j)$ correctly ordered & contains $x_0 .. x_j$

if WHILE-loop runs

$\Rightarrow$ since $i \geq 0$ as $j \geq 1$, we have $A[i] > key$.

L5-6 ensure that entry $A[i]$ is "shifted" to entry $A[i+1]$ as long as $A[i] > k$ & $i \geq 0$

that is in L7

we have for A: 

$\Rightarrow$ $A(0..i+1)$ & $A(i+1..j)$ is sorted:

$A(0) \leq A(1) \leq \dots \leq A(i+1) \leq \dots A(j)$

& contains $x_0 \dots x_j$. $\Rightarrow$ II holds.

Since the latter holds for each iteration of FOR-loop,

$A(0..n-1)$ is sorted & contains $x_0 .. x_{n-1}$ /

[(III) hold]

O, Ω, Θ - notation

▶ $f(n) = \frac{1}{2} n^2 + 3n$ in $\Omega(n^2)$, $O(n^2)$, $\Theta(n^2)$

$f(n) \in \Omega(n^2)$: must show exist pos. const c_1, n_0 s.t.

$$c_1 g(n) = c_1 \cdot n^2 \leq f(n) = \frac{1}{2} n^2 + 3n \quad \forall\, n \geq n_0$$

$$\Leftrightarrow \quad c_1 n^2 \leq \frac{1}{2} n^2 + 3n \quad | - \frac{1}{2} n^2$$

$$(c_1 - \tfrac{1}{2}) n^2 \leq 3n \quad | \cdot \tfrac{1}{n}$$

$$(c_1 - \tfrac{1}{2}) n \leq 3$$

holds e.g. for $c_1 = \frac{1}{3}$ & $n \geq n_0 = 1$

since $\underbrace{\left(\frac{1}{3} - \frac{1}{2}\right)}_{< 0} \cdot n < 0 \leq 3$

$f(n) \in O(n^2)$: must show exist pos. const c_2, n_0 s.t.

$$f(n) = \frac{1}{2} n^2 + 3n \leq c_2 n^2 \quad \forall \quad n \geq n_0$$

$$\Leftrightarrow \quad \left(\frac{1}{2} - c_2\right) n^2 \leq -3n \quad | \cdot \tfrac{1}{n^2}$$

$$\Leftrightarrow \quad \left(\frac{1}{2} - c_2\right) \leq -\frac{3}{n}$$

holds eg for $c_2 = 5$, $n \geq n_0 = 1$

since then $\left(\frac{1}{2} - 5\right) = -4.5 \leq -3 \leq -\frac{3}{n}$

$f(n) \in \Theta(n^2)$: choose $c_1 = \frac{1}{3}$, $c_2 = 5$, $n_0 = 1$

$f(n) = \frac{1}{2} n^2 + 3n$ $\quad \notin O(n)$, $\in \Omega(n)$

$\notin O(n)$: $\quad \frac{1}{2} n^2 + 3n \leq c \cdot n$ for some const. c, $n_0 \geq 0$ & all $n \geq n_0$?

$\frac{1}{2} n^2 \leq (c-3) \cdot n \quad | \cdot \frac{2}{n}$

$n \leq 2c - 6$

there is always an $n > 2c-6$ $\forall n \geq n_0 \geq 0$ since c is fixed ↯

$\in \Omega(n)$: $\quad \frac{1}{2} n^2 + 3n \geq cn$

$\frac{1}{2} n^2 + 3n \geq \frac{1}{2} n^2 \geq cn \quad | \cdot \frac{2}{n}$

$n > 2c$

choose $c = 1$, $n_0 = 2$

$\Rightarrow f(n) \geq cn \quad \forall \; n \geq n_0 = 2, \; c = 1.$

▶ $f(n) = \frac{1}{2}n^2 + 3n \quad \in O(n^3)$, $\notin \Omega(n^3)$

$\in O(n^3)$: $\quad \frac{1}{2}n^2 + 3n \leq cn^3$

$0 \leq cn^3 - \frac{1}{2}n^2 - 3n$

$= n(cn^2 - \frac{1}{2}n - 3) \quad n \geq n_0 \geq 0$

$\Rightarrow$ when is $cn^2 - \frac{1}{2}n - 3 = 0$?

$n_{1,2} = \frac{\frac{1}{2} \pm \sqrt{\frac{1}{4} + 4 \cdot 3 \cdot c}}{2c}$ for any $c > 0$, say $c = 1$

$\overset{c=1}{=} \frac{\frac{1}{2} \pm \sqrt{12{,}25}}{2} = \frac{0{,}5 \pm 3{,}5}{2}$

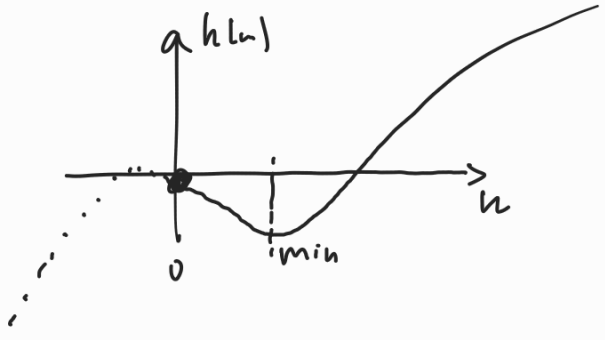
Since $n_0 \geq 0$ $\Rightarrow$ $n_0 = 2$ $\Rightarrow$ $f(n) \in O(n^3)$

$\notin \Omega(n^3)$: $\quad c \cdot n^3 \leq \frac{1}{2}n^2 + 3n \quad | \cdot \frac{1}{n^3}$

$\Leftrightarrow$ $h(n) = cn^3 - \frac{1}{2}n^2 - 3n \leq 0$

Sketch (no official proof, but idea)

$\forall c > 0$:



that is there is no n_0 st for all $n \geq n_0$ it holds $h(n) \leq 0$

$\Leftrightarrow cn^3 \leq \frac{1}{2}n^2 + 3n$.

- $2^{n+1} \in \Theta(2^n)$?

$2^{n+1} = 2 \cdot 2^n \Rightarrow 2 \cdot 2^n \leq 2^{n+1} \leq 2 \cdot 2^n$

$\Rightarrow$ choose $c = 2$, $n_0 = 1$

- $f(n) = n^2 (\sin(n))^2 + 50n \in \begin{matrix} O(n^2) \\ \Omega(n) \end{matrix}$

$\sin(n) \leq 1 \Rightarrow f(n) \leq \underbrace{n^2 + 50n \leq cn^2}$

$\Leftrightarrow \quad 50 \leq (c-1)n = c'n$

$50 \leq c' \cdot n \quad$ for $c' = 1$, $n = 50$

$\Rightarrow f(n) \in O(n^2)$

$n^2 (\sin(n))^2 \geq 0 \Rightarrow f(n) \geq 50n \quad \forall n \geq 1 \Rightarrow f(n) \in \Omega(n)$.

Transitivity: $f(n) \in O(g(n))$ & $g(n) \in O(h(n))$
$\Rightarrow f(n) \in O(h(n))$.

proof: $f(n) \in O(g(n)) \Rightarrow \exists c, n_0$ s.t. $f(n) \le c \cdot g(n)$ $\forall n \ge n_0$ I
$g(n) \in O(h(n)) \Rightarrow \exists c', n_0'$ s.t. $g(n) \le c' \cdot h(n)$ $\forall n \ge n_0'$ II

let $N_0 = \max\{n_0, n_0'\}$, $e = \max\{c, c'\}$

I: $f(n) \le c \cdot g(n) \le e\, g(n)$ $\forall$ $n \ge N_0 \ge n_0$

II: $g(n) \le c' \cdot h(n) \le e \cdot h(n)$ $\forall$ $n \ge N_0 \ge n_0'$

I & II: $f(n) \le e\, g(n) \le e(e\, h(n)) = e^2 h(n) = \underbrace{K}_{\text{pos. const}} h(n)$

$\forall n \ge N_0$

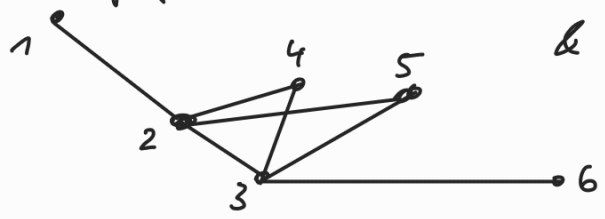
$\Rightarrow f(n) \in O(h(n))$.

(similar arguments for Ω & Θ)

$\square$

BASICS: Graphs & trees (part 1)

A graph $G = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6\}$ & $E = \{\{1,2\}, \{2,3\}, \{2,4\}, \{2,5\}, \{3,4\}, \{3,5\}, \{3,6\}\}$



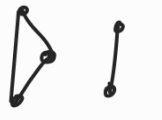
(connected)

path $P = (1,2,4,3,2,5)$ connecting 1 & 5 of length 5 [not simple]

simple path e.g. $P = (1,2,5)$ length 2 (# edges)

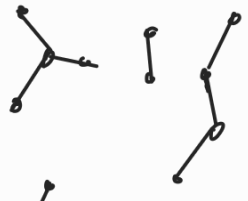
cycle $P = (2,3,4,2)$

6 not connected

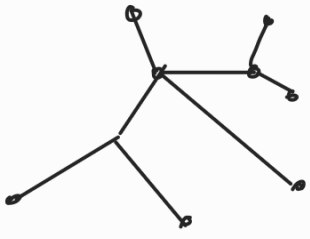


← contain cycles

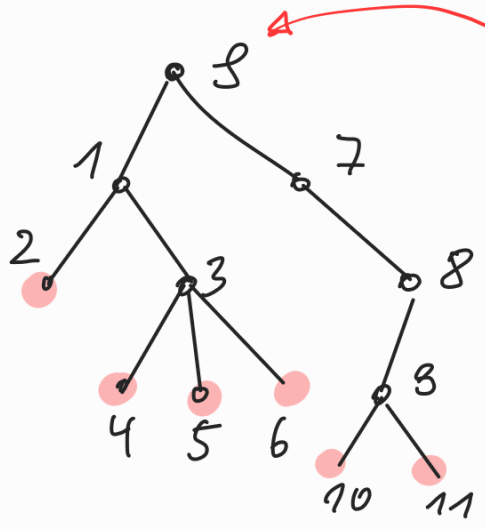
acyclic / forest



tree



rooted tree:



root usually drawn on top!

$5 \leq_T 1$

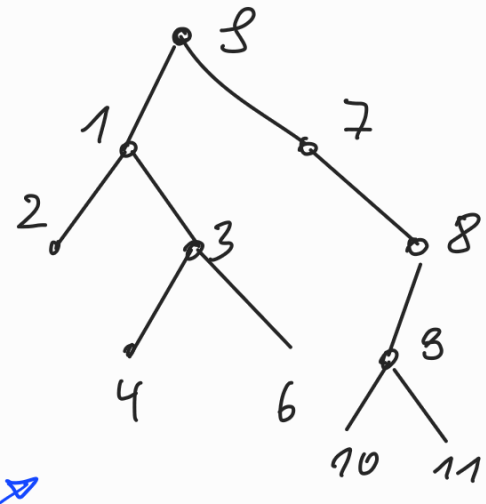
1 is parent of 3, 3 is child of 1

leaves highlighted

2 & 3 are siblings

binary tree

T

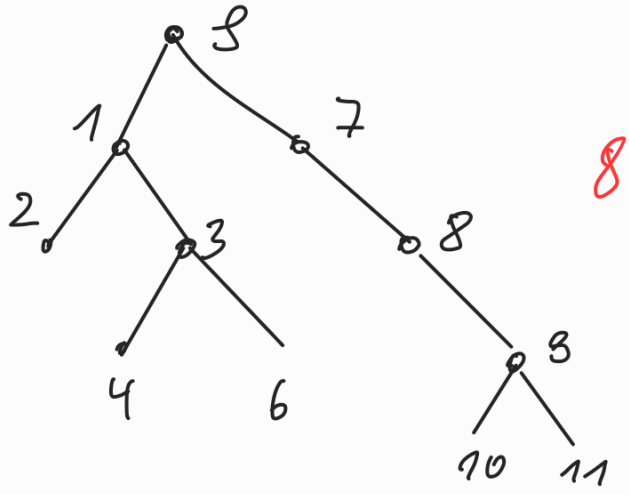


← 8 has a left child (9) but no right child.

drawing implies order on children, e.g.,

"4 left of 6"
"11 right of 10"

T'



8 has right child but no left child

⇒ T & T' are different trees!!