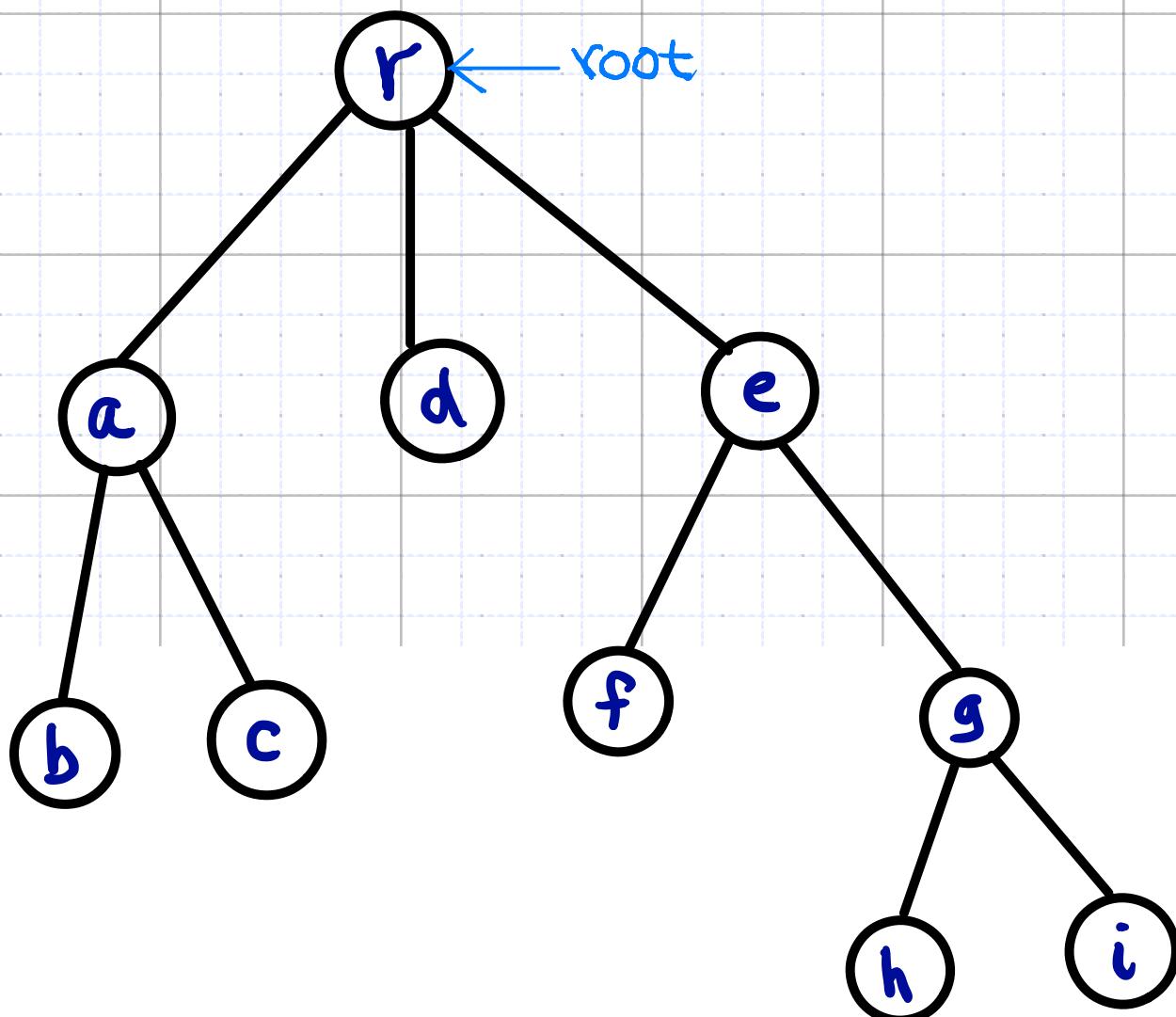


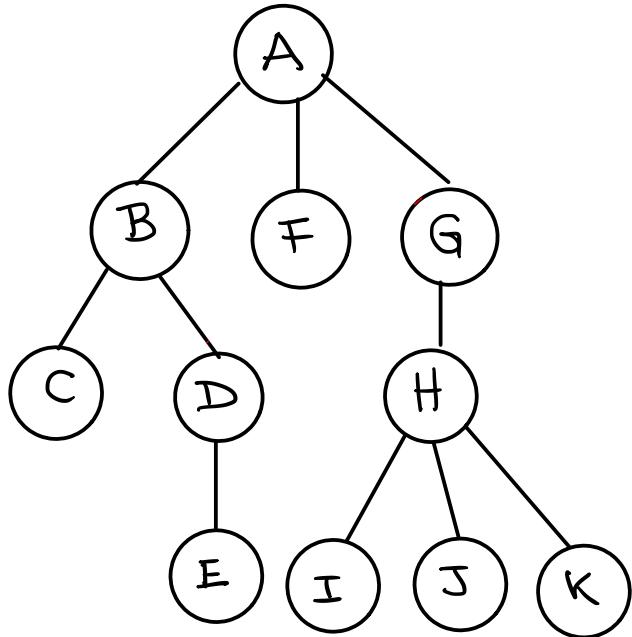
Tree Traversals

A tree is specified by only indicating its root.

A tree traversal visits the nodes of a tree in a systematic manner



Preorder Traversal



Algorithm preorder (r)

In: root r of a tree

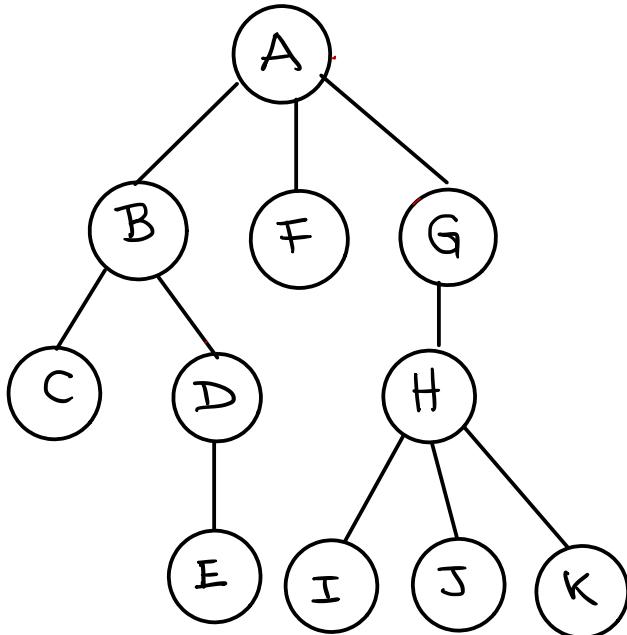
Out: nodes in preorder
visit(r)

For each child c of r do
 preorder(c)

Preorder traversal of this tree:

A B C D E F G H I J K

Postorder Traversal



Algorithm postorder (r)

In: root r of a tree

Out: nodes in postorder

For each child c of r do

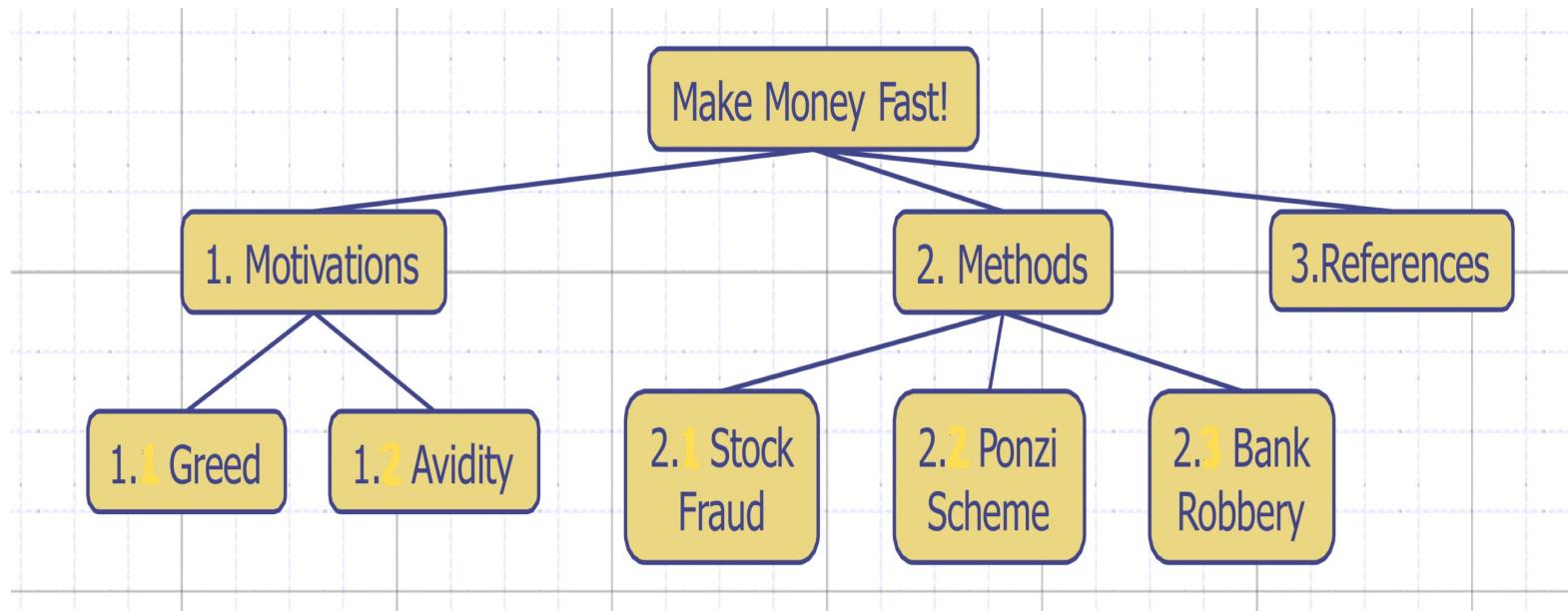
postorder(c)

visit(r)

Postorder traversal of this tree:

C E D B F I J K H G A

Print the Table of Contents of a Book



Make Money Fast!

1. Motivations

- 1. Greed
- 2. Avidity

2. Methods

- 1. Stock Fraud
- 2. Ponzi Scheme
- 3. Bank Robbery

3. References

Print the Table of Contents of a Book

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

TOC(c, indentation+1)

Execution of the Algorithm

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

TOC(c, indentation+1)

[addr 2]

Algorithm main()

TOC(A,0)

[addr 1]

r=F i=2
indentation=2 c=
ret addr = addr 2

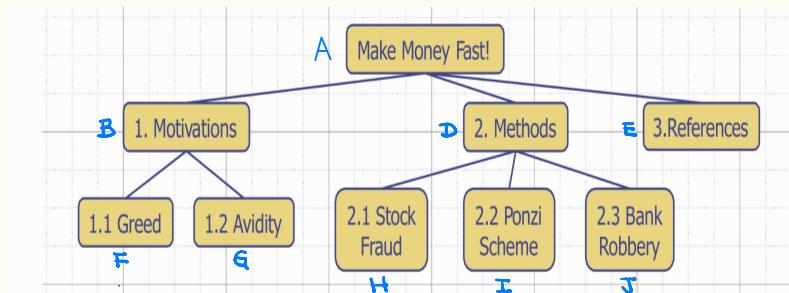
r=B i=1
indentation=1 c=F
ret addr = addr 2

r=A i=0
indentation=0 c=B
ret addr = addr 1

ret addr = 0S

} activation record of Toc
} activation record of main

Execution stack



Output

Make Money Fast!

1.Motivations
1.1 Greed

Execution of the Algorithm

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

TOC(c, indentation+1)

[addr 2]

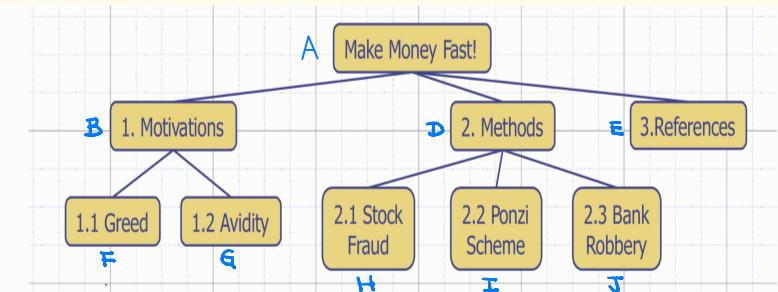
Algorithm main()

TOC(A,0)

[addr 1]

r=B	i=1
indentation =1	c=G
ret addr = addr 2	
r=A	i=
indentation =0	c=B
ret addr = addr 1	
	ret addr = 05

activation
record
popped



Output

Make Money Fast!

1.Motivations
1.1 Greed

Execution stack

Execution of the Algorithm

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

TOC(c, indentation+1)

[addr 2]

Algorithm main()

TOC(A,0)

[addr 1]

r=G i=2
indentation=2 c=
ret addr = addr 2

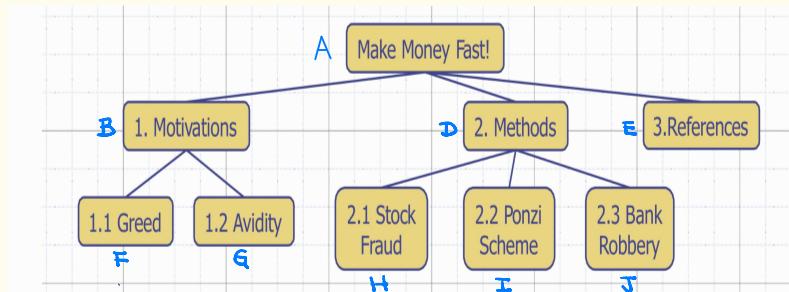
r=B i=1
indentation=1 c=F
ret addr = addr 2

r=A i=0
indentation=0 c=B
ret addr = addr 1

ret addr = 0S

} activation record of Toc
} activation record of main

Execution stack



Output

Make Money Fast!

1.Motivations
1.1 Greed
1.2 Avidity

Execution of the Algorithm

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

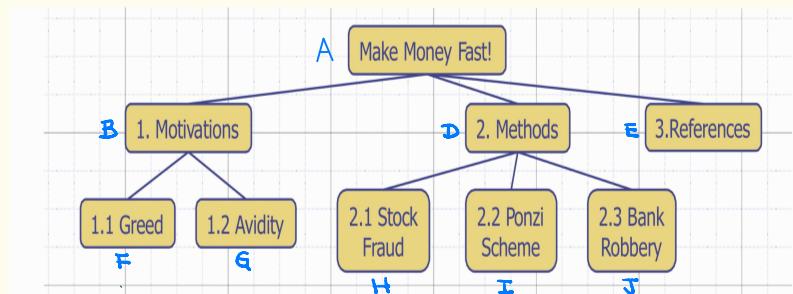
TOC(c, indentation+1)

[addr 2]

Algorithm main()

TOC(A,0)

[addr 1]



Output

Make Money Fast!

1.Motivations

1.1 Greed

1.2 Avidity

r=A
indentation = 0 i=
ret addr = addr 1

ret addr = 0S

} activation
record of
TOC

} activation
record of
main

Execution stack

Execution of the Algorithm

Algorithm TOC (r, indentation)

In: root r of a tree representing the table of contents of a book,
and integer indentation (of value zero in the first call)

Out: {print the table of contents properly indented}

for i \leftarrow 1 to indentation do print(" ")

print content of r

for each child c of r do

TOC(c, indentation+1)

[addr 2]

Complete the execution
of the algorithm

r=H i=2
indentation=2 c=
 ret addr = addr 1

r=D i=1
indentation=1 c=H
 ret addr = addr 1

r=A i=0
indentation=0 c=D
 ret addr = addr 1

ret addr = 0S

} activation
record of
TOC

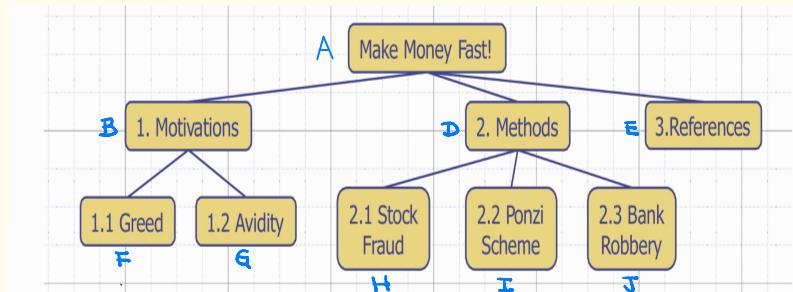
} activation
record of
main

Execution stack

Algorithm main()

TOC(A,0)

[addr 1]



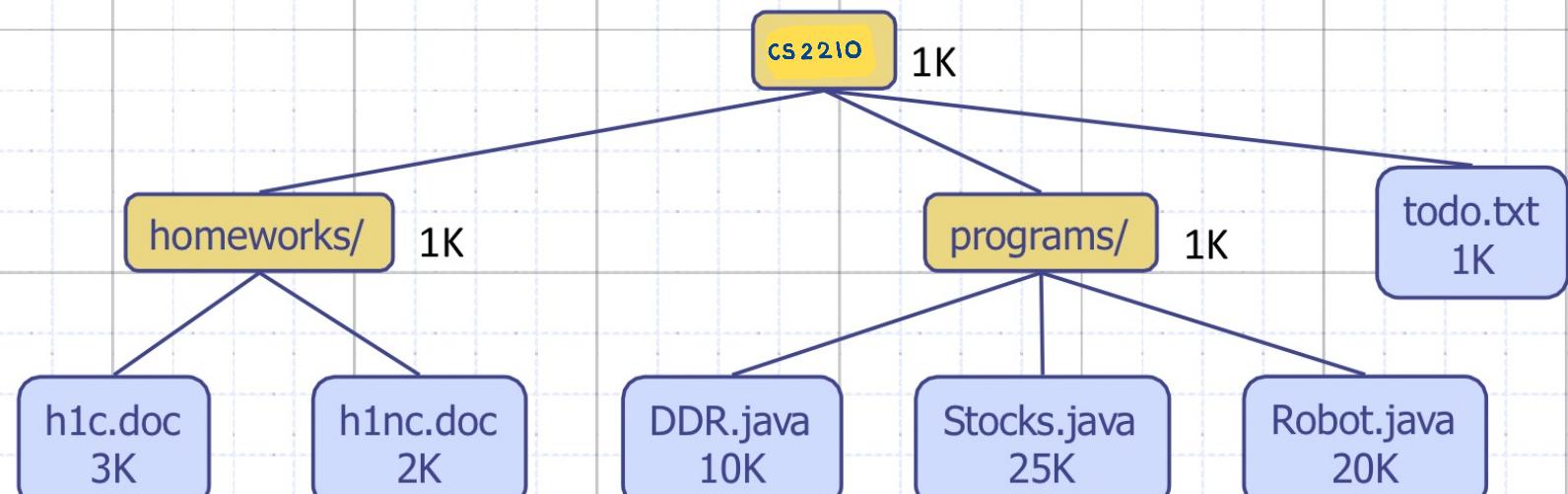
Output

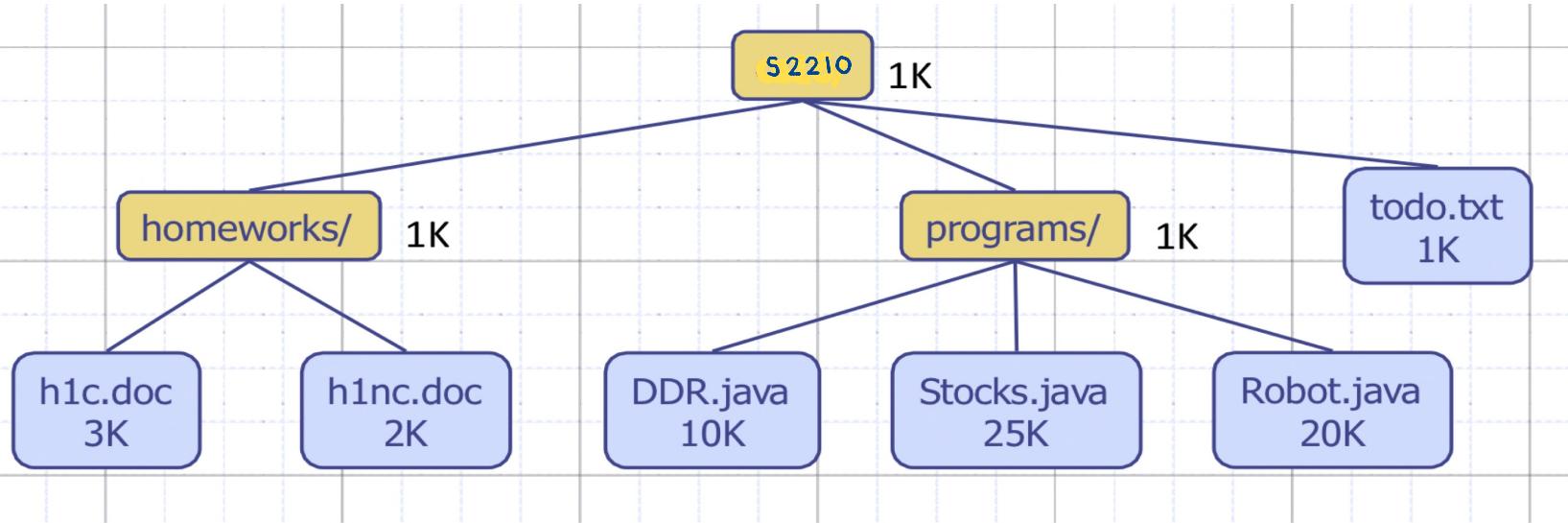
Make Money Fast!

1. Motivations
 1.1 Greed
 1.2 Avidity
2. Methods
 2.1 Stock Fraud

Application

Compute space used by the files in a directory and its subdirectories





Algorithm diskSpace(r)

In: root r of a file system tree

Out: Total space used by the file system

```

 $c_1 \left\{ \begin{array}{l} \text{sum} \leftarrow 0 \\ \text{for each child } c \text{ of } r \text{ do} \\ \quad \text{sum} \leftarrow \text{diskSpace}(c) + \text{sum} \end{array} \right. \quad \# \text{iterations} = \text{degree}(r)$ 
 $\left. \begin{array}{l} \text{sum} \leftarrow \text{sum} + r.\text{size} \\ \text{return sum} \end{array} \right\} c_2$ 

```

Operations per call

$$c_1 + c_2 \times \text{degree}(r)$$

Number of Calls

One per node (n total)

Total number of operations

$$\begin{aligned} f(n) &= \sum_{\text{nodes } u} (c_1 + c_2 \times \text{degree}(u)) \\ &= \sum_{\text{nodes } u} c_1 + c_2 \sum_{\text{nodes } u} \text{degree}(u) \end{aligned}$$

$$= c_1 n + c_2 (n-1) \text{ is } O(n)$$