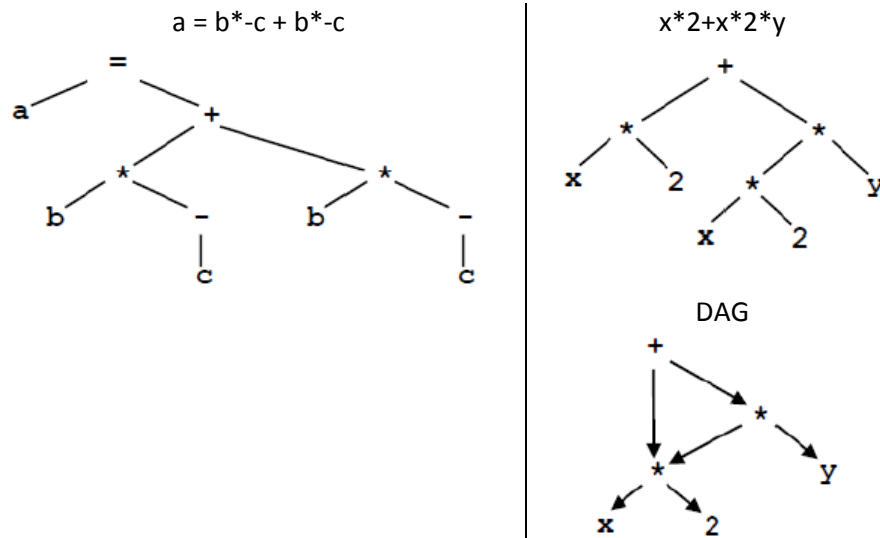


IR Taxonomy:

Three organizational categories:

- Graphical IRs encode the compiler's knowledge in a graph.
- Linear IRs resemble pseudo-code for some abstract machine.
- Hybrid IRs combine elements of both graphical (structural) and linear IRs.

Graphical IRs:



Linear IRs:

- Stack Machine Code
- Three-Address Code ($x \leftarrow y \text{ op } z$)

Example: $x - 2 * y$

Stack Machine	Three-Address
Push 2	$t1 \leftarrow 2$
Push y	$t2 \leftarrow y$
Multiple	$t3 \leftarrow t1 \times t2$
Push x	$t1 \leftarrow x$
Subtract	$t2 \leftarrow t1 - t3$

Representing Linear Codes:

Target	op	Arg1	Arg2
t1	\leftarrow	2	
t2	\leftarrow	y	
t3	\times	t1	t2
t1	\leftarrow	x	
t2	-	t1	t3

Simple Single-Dimensional Array of Quadruples

```

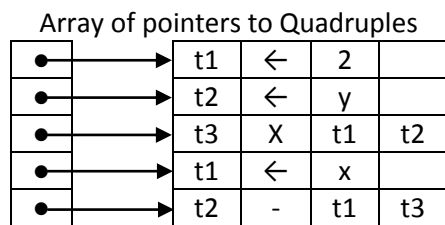
Class Q {
    ... target;
    ... op;
    ... arg1;
    ... arg2;
};

```

Q IR[n];

Index

0	t1	←	2	
1	t2	←	y	
2	t3	X	t1	t2
3	t1	←	x	
4	t2	-	t1	t3

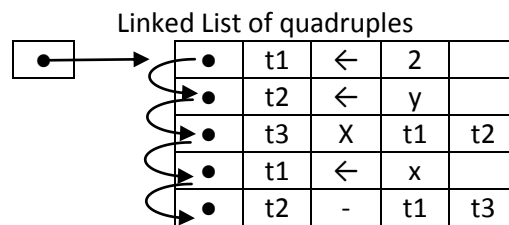


```

Class Q {
    ... target;
    ... op;
    ... arg1;
    ... arg2;
};

```

TASK !!!



```

Class Q {
    ... target;
    ... op;
    ... arg1;
    ... arg2;
    Q *next;
};

Q *first;

```

Example:

Code	Three-Address Code	Linear IR				
<pre> if c<d then x = y + z; else x = y - z; </pre>	<pre> if c<d then goto L1 goto L2 L1: x = y + z goto L3 L2: x = y - z L3: nop </pre>	Label	Target	op	Arg1	Arg2
			L1	If_lt	c	d
			L2	goto		
		L1	x	+	y	z
			L3	goto		
		L2	x	-	y	z
		L3		nop		