

# ANALYSIS OF ALGORITHMS

---

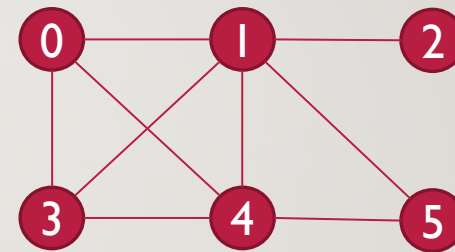
LECTURE 5 : GRAPHS



# WHY?

---

- What's the point of using graphs?
- Map colouring problem
- Now read sections 3.1 and 3.2

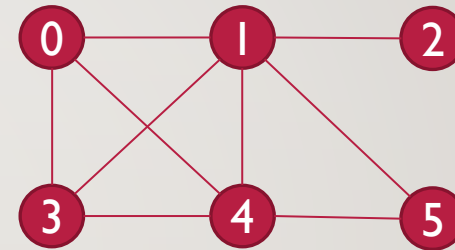


# REPRESENTING GRAPHS ON A COMPUTER

---

- Adjacency matrix

	0	1	2	3	4	5
0	-	T	F	T	T	F
1	T	-	T	T	T	T
2	F	T	-	F	F	F
3	T	T	F	-	T	F
4	T	T	F	T	-	T
5	F	T	F	F	T	-

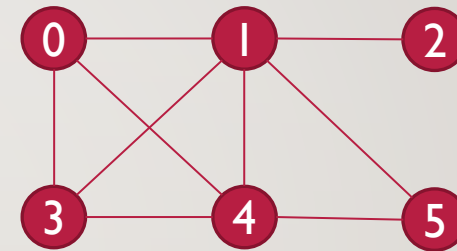


# REPRESENTING GRAPHS ON A COMPUTER

---

- Adjacency list

0	1,3,4
1	0,2,3,4,5
2	1
3	0,1,4
4	0,1,3,5
5	1,4



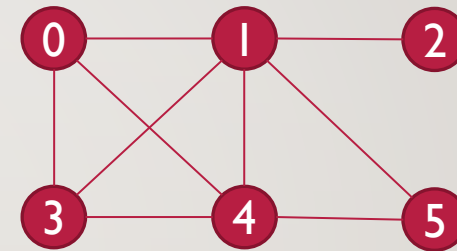
# REPRESENTING GRAPHS ON A COMPUTER

- Adjacency list

0	1,3,4
1	0,2,3,4,5
2	1
3	0,1,4
4	0,1,3,5
5	1,4

- Adjacency matrix

	0	1	2	3	4	5
0	-	T	F	T	T	F
1	T	-	T	T	T	T
2	F	T	-	F	F	F
3	T	T	F	-	T	F
4	T	T	F	T	-	T
5	F	T	F	F	T	-





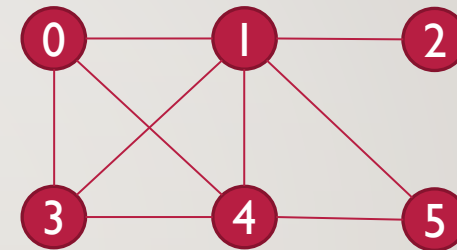
# REPRESENTING GRAPHS ON A COMPUTER

- Adjacency list

0	1,3,4
1	0,2,3,4,5
2	1
3	0,1,4
4	0,1,3,5
5	1,4

- Adjacency matrix

	0	1	2	3	4	5
0	-	T	F	T	T	F
1	T	-	T	T	T	T
2	F	T	-	F	F	F
3	T	T	F	-	T	F
4	T	T	F	T	-	T
5	F	T	F	F	T	-



Why do both of these representations exist?

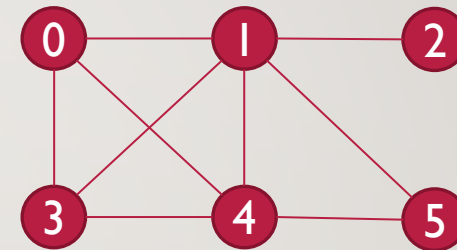
# REPRESENTING GRAPHS ON A COMPUTER

- Adjacency list

0	1,3,4
1	0,2,3,4,5
2	1
3	0,1,4
4	0,1,3,5
5	1,4

- Adjacency matrix

	0	1	2	3	4	5
0	-	T	F	T	T	F
1	T	-	T	T	T	T
2	F	T	-	F	F	F
3	T	T	F	-	T	F
4	T	T	F	T	-	T
5	F	T	F	F	T	-



- Consider the cost of checking whether two items are adjacent
- In an adjacency matrix, we just look up the item and see if true or false –  $\Theta(1)$
- In an adjacency list, we do a linear search in one vertex's list –  $\Theta(n)$
- Adjacency matrix WINS!

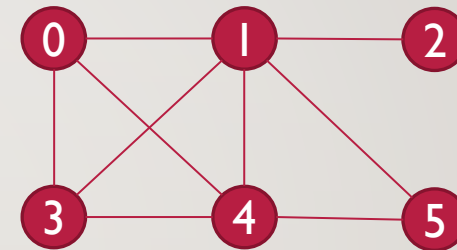
# REPRESENTING GRAPHS ON A COMPUTER

- Adjacency list

0	1,3,4
1	0,2,3,4,5
2	1
3	0,1,4
4	0,1,3,5
5	1,4

- Adjacency matrix

	0	1	2	3	4	5
0	-	T	F	T	T	F
1	T	-	T	T	T	T
2	F	T	-	F	F	F
3	T	T	F	-	T	F
4	T	T	F	T	-	T
5	F	T	F	F	T	-



- Consider the cost of checking the degree of a vertex
- In an adjacency matrix, we count the number of Trues in a row –  $\Theta(n)$
- In an adjacency list, we check the size of the vertex's list –  $\Theta(1)$
- Adjacency list WINS!