Pre-computing Paths

To Search Or Not To Search?

- Pre-computed optimal paths can be stored for some or all of a navigation graph
    - Benefit: little or search required during play
    - Costs: **memory** + precomputation
- Store as 'next node' look-up table
    - Given current and goal node, returns next node in path
    - Finding an N node path requires only N look ups
    - Table requires O(nodes^2) memory

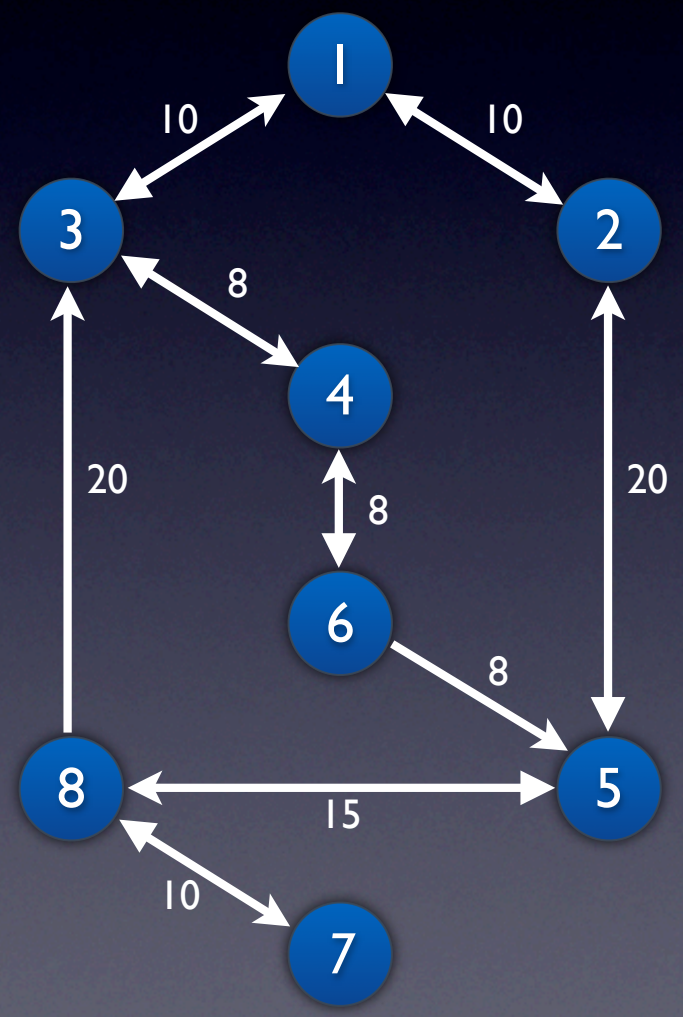
Pre-computing Paths

A Next Node Table

Goal node

Current node

	1	2	3	4	5	6	7	8
1	-	2	3	3	2	3	2	2
2	1	-	1	1	5	1	5	5
3	1	1	-	4	4	4	4	4
4	3	3	3	-	6	6	6	6
5	2	2	8	8	-	8	8	8
6	4	5	4	4	5	-	5	5
7	8	8	8	8	8	8	-	8
8	3	5	3	3	5	3	7	-

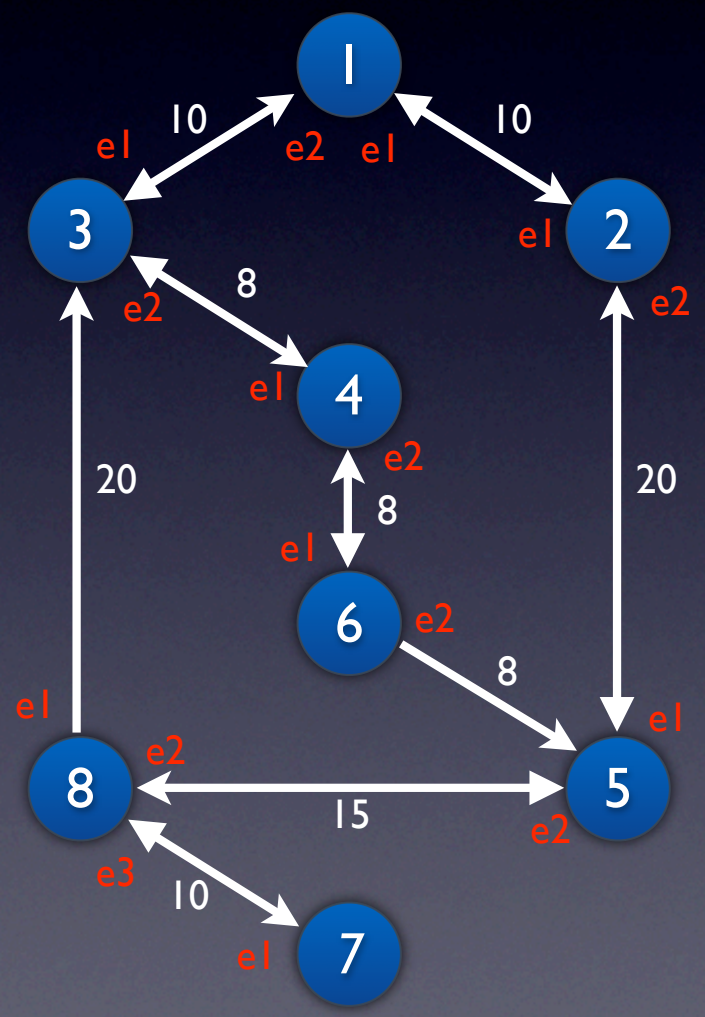


Pre-computing Paths

A Next Edge Table

Goal node

Current node

	1	2	3	4	5	6	7	8
1	-	e1	e2	e2	e1	e2	e1	e1
2	e1	-	e1	e1	e2	e1	e2	e2
3	e1	e1	-	e2	e2	e2	e2	e2
4	e1	e1	e1	-	e2	e2	e2	e2
5	e1	e1	e2	e2	-	e2	e2	e2
6	e1	e2	e1	e1	e2	-	e2	e2
7	e1	e1	e1	e1	e1	e1	-	e1
8	e1	e2	e1	e1	e2	e1	e3	-

Storing edge indices ("take the 3rd edge") rather than node labels requires less memory, as the maximum degree is typically much less than the number of nodes