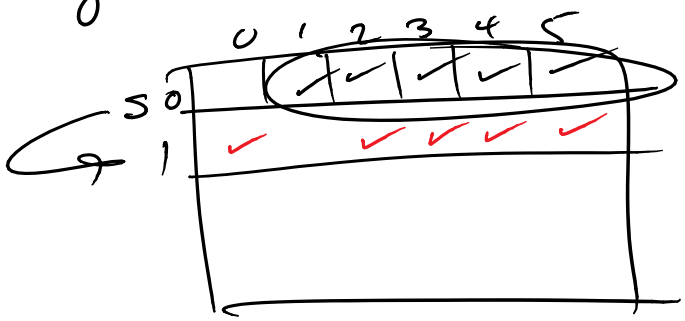
```
int main()
{
    DGraph G;
    G.load();
}
```

10% extra - Sunday

Wednesday 11:59 PM final due date

Dijkstra's Algorithm

Single Source - Shortest Path

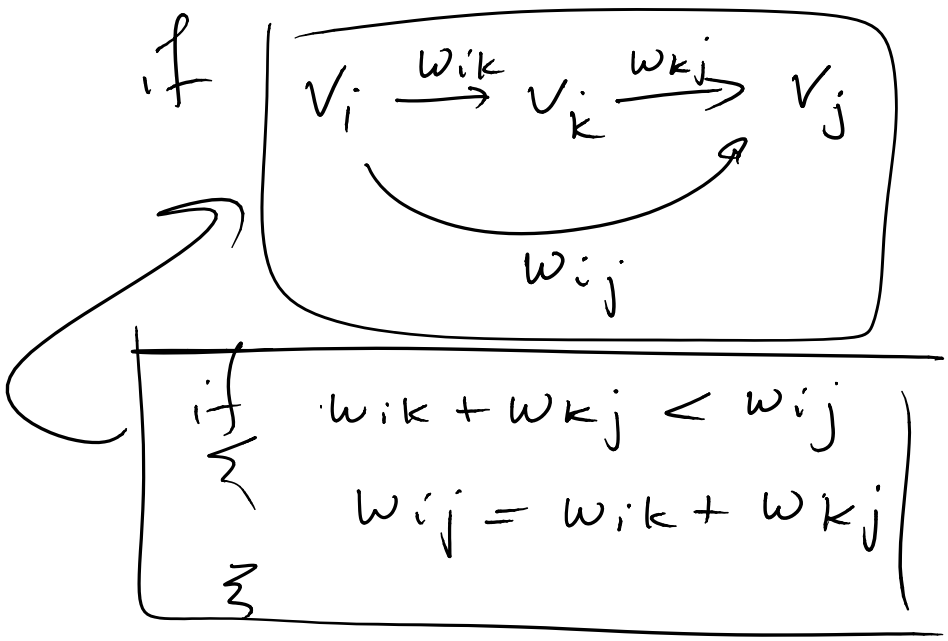


Floyd's Algorithm

All Pairs - Shortest Path

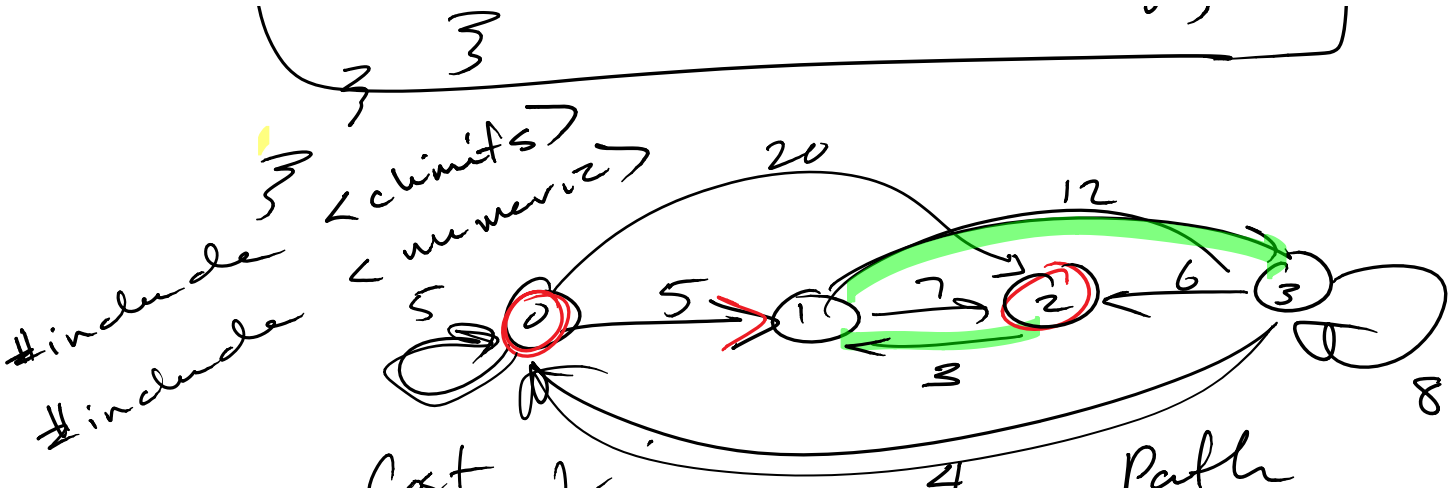
① Given and adjacency matrix, A ← weighted

② For all vertices V_i, V_j, V_k,



$G = \langle V, E \rangle$ Adjacency A

```
for(i=0; i<|V|; i++) A[i][i] = 0;
for (int i=0; i<|V|; i++)
{ for (int j=0; j<|V|; j++)
  { for( int k=0; k<|V|; k++)
    { if ( A[i][k]+A[k][j]
           < A[i][j])
        A[i][j] = A[i][k]+
                  A[k][j];
    }
  }
}
```



Cost

	0	1	2	3
0	0	5	12	17
1	16	0	17	12
2	19	3	10	15
3	4	9	6	0

Path

	0	1	2	3
0	-1	-1	1	1
1	3	-1	-1	-1
2	1	-1	-1	1
3	-1	2	-1	-1

2 → 1 → 0

```
void ppath (vector<vector<int>> path, i, j)
{
}
```

cost

5 pts

no path

1 → 2 → 5 → 0 (20)