CS 225

Data Structures

April 30 – Floyd-Warshall's Algorithm

Wade Fagen-Ulmschneider

Floyd-Warshall Algorithm

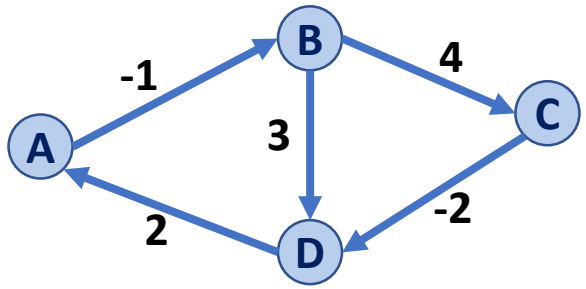
Floyd-Warshall's Algorithm is an alterative to Dijkstra in the presence of negative-weight edges (not negative weight cycles).

```
    FloydWarshall(G):
 6    Let d be a adj. matrix initialized to +inf
 7    foreach (Vertex v : G):
 8      d[v][v] = 0
 9    foreach (Edge (u, v) : G):
10      d[u][v] = cost(u, v)
11
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex w : G):
15          if d[u, v] > d[u, w] + d[w, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

Floyd-Warshall Algorithm

```
    FloydWarshall(G):
 6    Let d be a adj. matrix initialized to +inf
 7    foreach (Vertex v : G):
 8      d[v][v] = 0
 9    foreach (Edge (u, v) : G):
10      d[u][v] = cost(u, v)
11
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex w : G):
15          if d[u, v] > d[u, w] + d[w, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

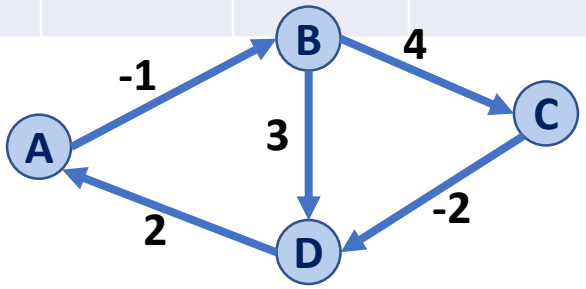
	A	B	C	D
A				
B				
C				
D				



Floyd-Warshall Algorithm

```
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex k : G):
15          if d[u, v] > d[u, k] + d[k, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

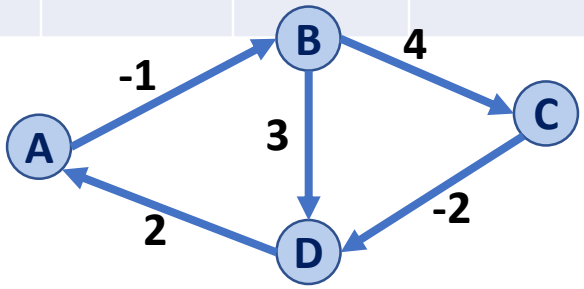
	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	∞	∞	0



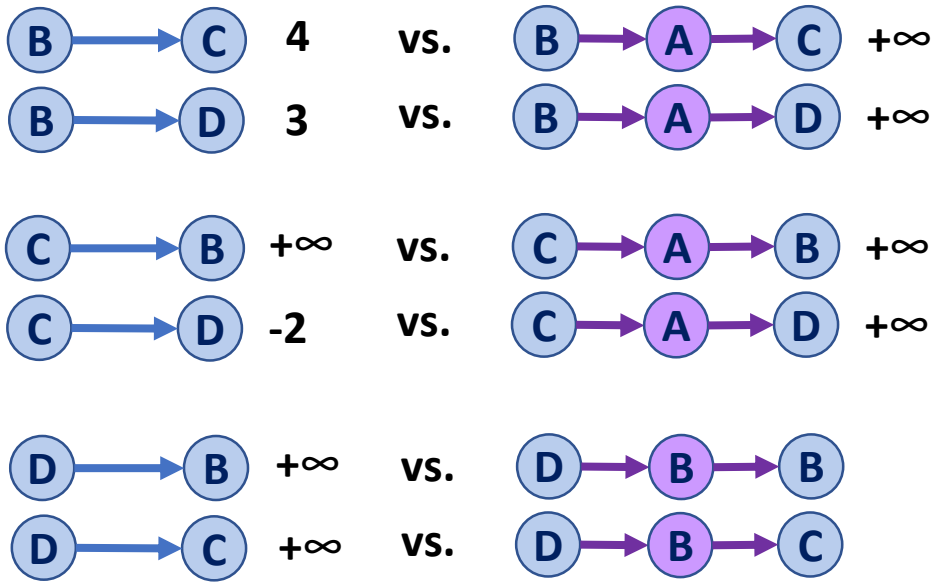
Floyd-Warshall Algorithm

```
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex k : G):
15          if d[u, v] > d[u, k] + d[k, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	∞	∞	0



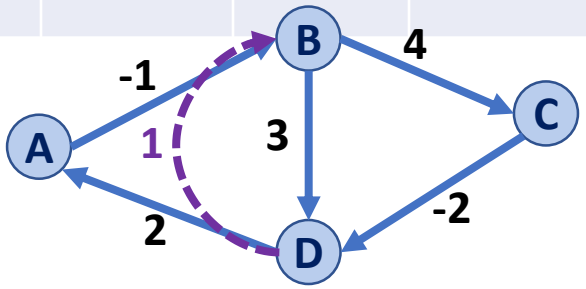
Let us consider k=A:



Floyd-Warshall Algorithm

```
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex k : G):
15          if d[u, v] > d[u, k] + d[k, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

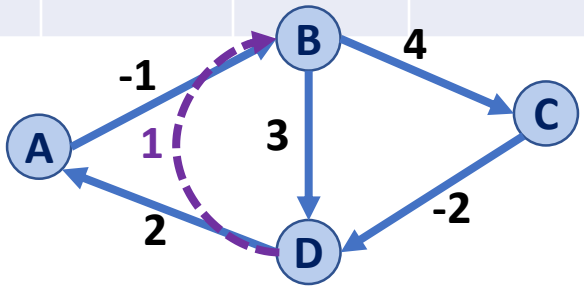
	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	1	∞	0



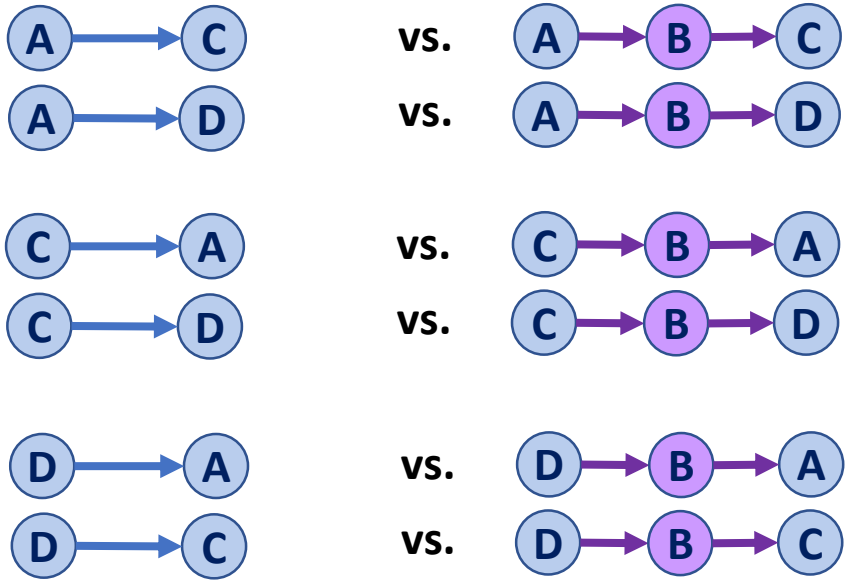
Floyd-Warshall Algorithm

```
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex k : G):
15          if d[u, v] > d[u, k] + d[k, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	1	∞	0



Let us consider k=B:



Floyd-Warshall Algorithm

Running Time?

```
    FloydWarshall(G):
 6    Let d be a adj. matrix initialized to +inf
 7    foreach (Vertex v : G):
 8      d[v][v] = 0
 9    foreach (Edge (u, v) : G):
10      d[u][v] = cost(u, v)
11
12    foreach (Vertex u : G):
13      foreach (Vertex v : G):
14        foreach (Vertex w : G):
15          if d[u, v] > d[u, w] + d[w, v]:
16            d[u, v] = d[u, w] + d[w, v]
```

Final Exam Review Session

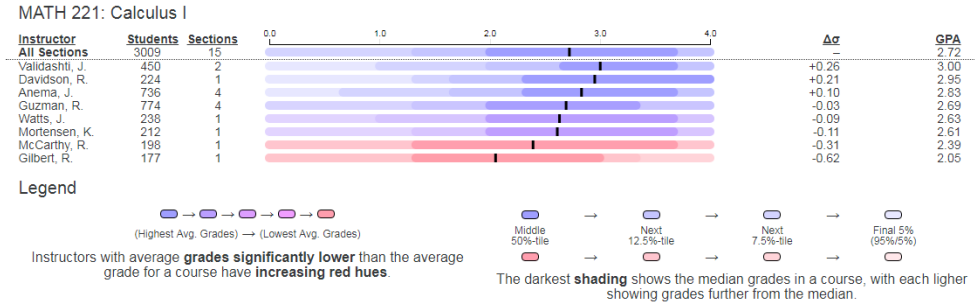
- Implementations
  - Edge List
  - Adjacency Matrix
  - Adjacency List
- Traversals
  - Breadth First
  - Depth First
- Minimum Spanning Tree
  - Kruskal's Algorithm
  - Prim's Algorithm
- Shortest Path
  - Dijkstra's Algorithm
  - Floyd-Warshall's Algorithm

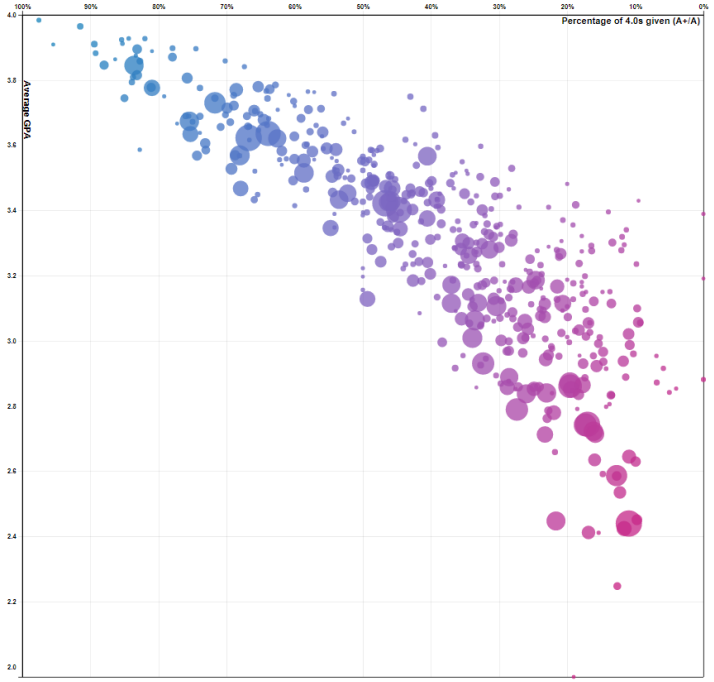
...and this is just the beginning. The journey continues to CS 374!

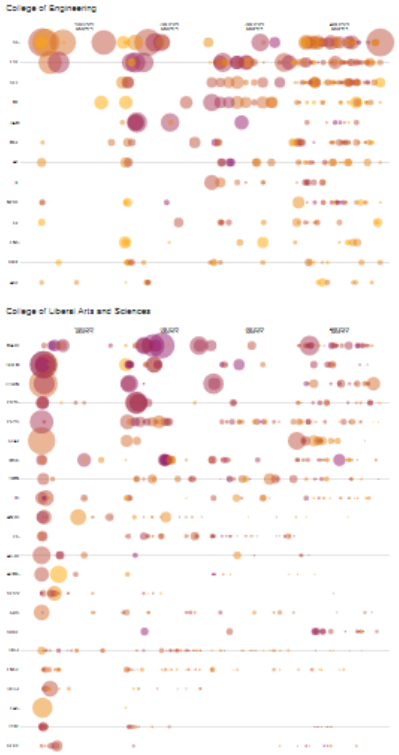
My Passion: Data Discovery

http://waf.cs.illinois.edu/discovery/

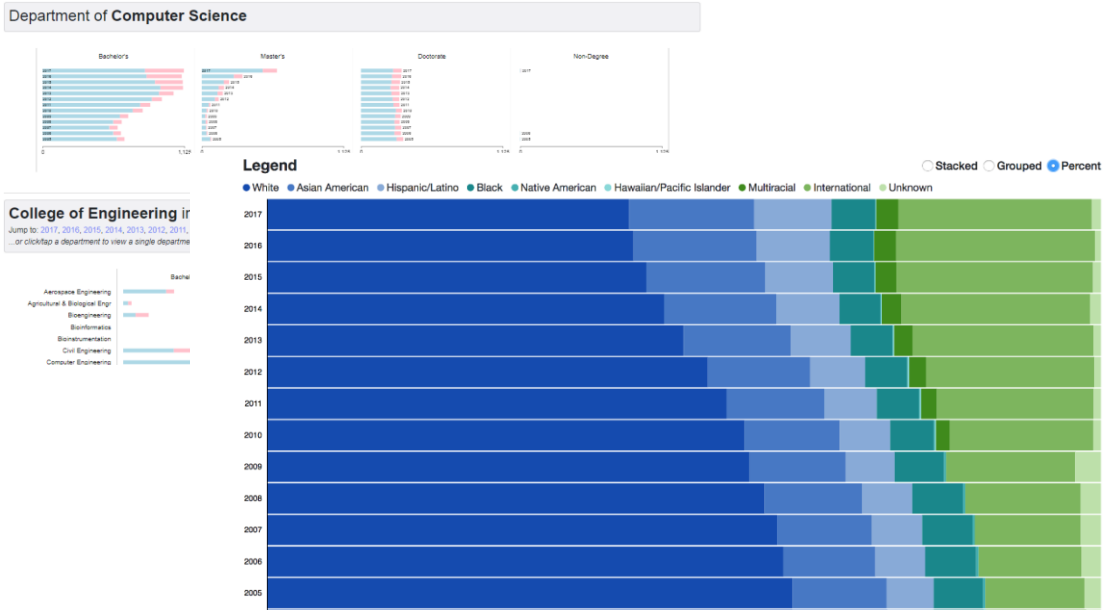
GPAs at Illinois:



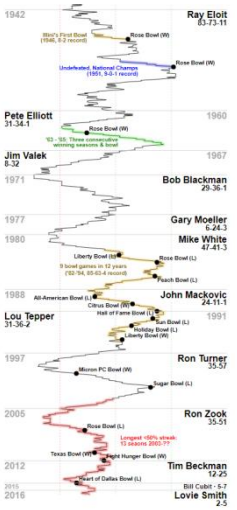




Diversity at Illinois:



And others:



Final Exam Review Session

- CS 225 will be doing an in-lecture final exam review session during Wednesday's lecture.

ICES Forms

My promise: I will read the back of every ICES form.
Please take the time to give feedback on the course.

Thanks for an amazing semester!