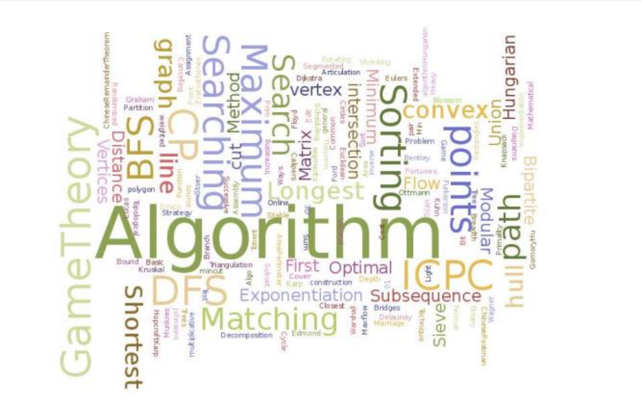
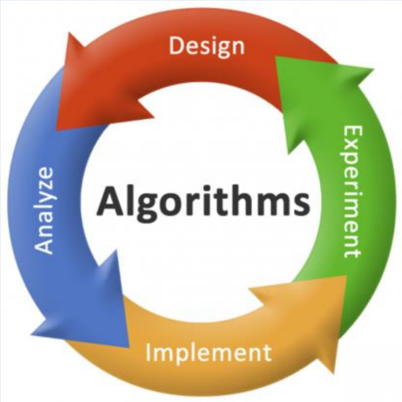
Graph Algorithms
Dijkstra's Algorithm

COP 3503
Fall 2021
Department of Computer Science
University of Central Florida
Dr. Steinberg



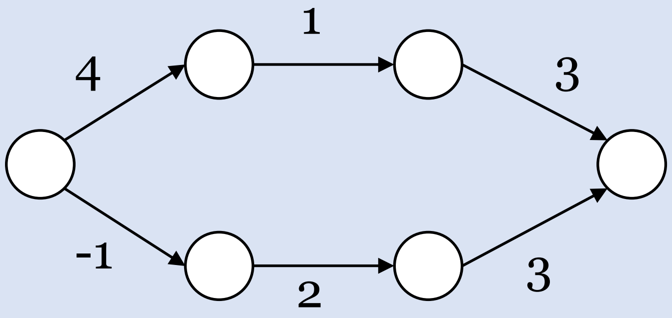


Single-Source Shortest Path

- The Problem Definition
  - Input: a directed weighted G(V,E) and a source vertex s
  - Output: The shortest path from s to destination vertex v
- Weight of a path

$$p = < v_0, v_1, v_2, \ldots v_k >$$

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$



- A shortest path has at most |V| vertices and at most |V| - 1 edges
- Some algorithms allow negative weight edges and others do not.

Dijkstra's Algorithm

- Solve the single-source shortest path problem on a weighted directed graph $G(V, E)$
- All edges on the directed graph MUST be nonnegative weights!
  - $w\ (u, v) \geq 0$ for each edge $(u, v) \in E$
- Dijkstra's Algorithm maintains a set S of vertices whose final shortest-path weights from the source have already been determined.
- The algorithm repeats and selects a vertex $u \in V - S$ with the minimum shortest path estimate.
  - Add u to S and relax all edges leaving u.

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Running Time Analysis of Dijkstra

- There are two cases to observe based on implementation
- Case 1
  - Q is implemented using a minimum priority queue
  - Each element has a key which is given by v.d
  - Implemented using a min-heap
  - Operations in queue take O(log n)
  - Relax (v,u,w) using queue takes O(logV)
- Overall the running time is $O\big((V+E)\lg V\big)$

Running Time Analysis of Dijkstra cont.

- Case 2
  - Array implementation
  - Sparse Graph: $|E| = O(V)$
  - Dense Graph: $|E| = O(V^2)$
  - RT = $O(V^2)$ in the worst case scenario

Example