Algorithms and Data Structures II

Lecture 5:

Shortest Path Problems

https://elms.u-aizu.ac.jp

Assumption:

- Let $G(V, E, W)$ be a weighted graph with non-negative edge distances (or costs).
- For edge (u, v), the distance of (u, v) is denoted by $d(u, v)$.
- The distance of a path P, denoted by $d(P)$, is the sum of the distances of edges in the path.
- For two nodes u and v in G, the shortest path from u to v is the path P such that $d(P) = \min\{d(Q) | Q \text{ is a path from } u \text{ to } v\}$.

Property on shortest path

- Let $u \to v \to w$ be a shortest path from u to w.
- Then $u \to v$ is a shortest path from u to v and $v \to w$ is a shortest path from v to w.

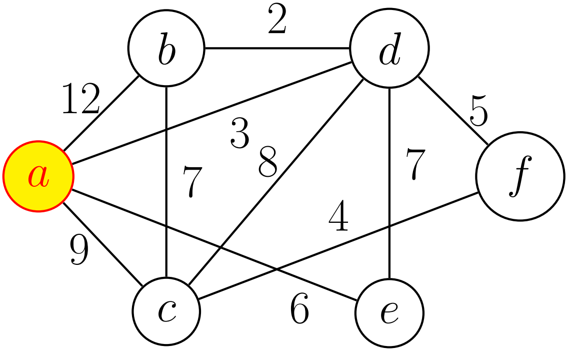
Shortest path problems

- Single source shortest path problem and all pairs shortest path problem are most important shortest path problems.
- Single source shortest path problem is the problem of finding the shortest paths from a specific vertex s, called source, to all other vertices of G (connected graph).
- All pairs shortest path problem finds the shortest paths between every pair of vertices in G (sources are all vertices in G).

Single Source Shortest path problem

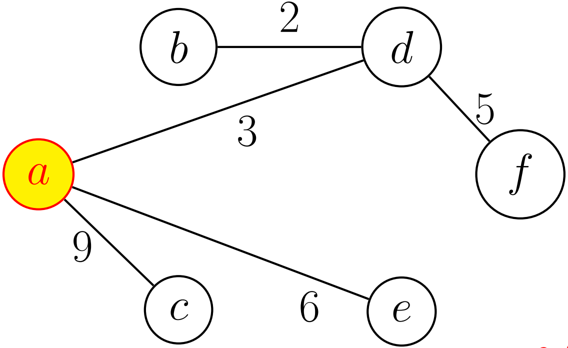
- Let $G(V, E, W)$ be a weighted graph with non-negative edge distances, and let s be a vertex of G from which every vertex of G can be reached.
- Then there exists a spanning tree T of G, rooted as s, which contains a shortest path from s to every vertex of G.
- Such a tree is called shortest path spanning tree.

Weighted Graph and its Shortest Path Spanning Tree



A Weighted Graph G

A Shortest Path Spanning Tree of G



Dijkstra's Algorithm

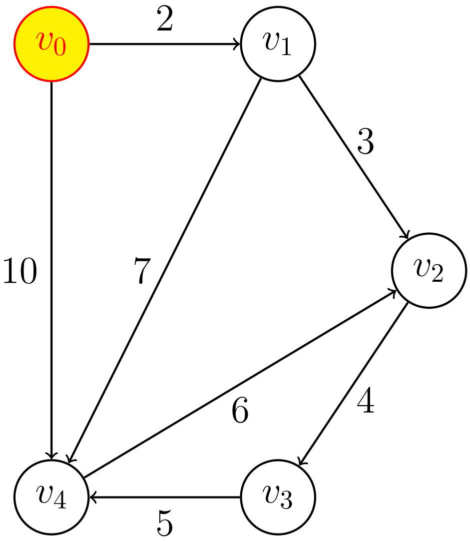
- In the algorithm, graph $G(V, E, W)$ is represented by distance matrix D.

  /* s is the source vertex and S is the partial solution set such that the shortest paths from s to each vertex in S lies wholly in S. For each vertex $v \in V - S$, $d[v]$ contains the distance of current shortest path from s to v passing only through vertices of S.*/

Dijkstra's Algorithm(contd.)

```
S = {s};  d[s] = 0;
for (v ∈ V − S)  d[v] = D[s, v];
while (S ≠ V) {
   Choose a vertex w in V − S such that
         d[w] is a minimum;
   add w to S;
   for (v ∈ V − S)  d[v] = min{d[v], d[w] + D[w, v]};
}
```

Example

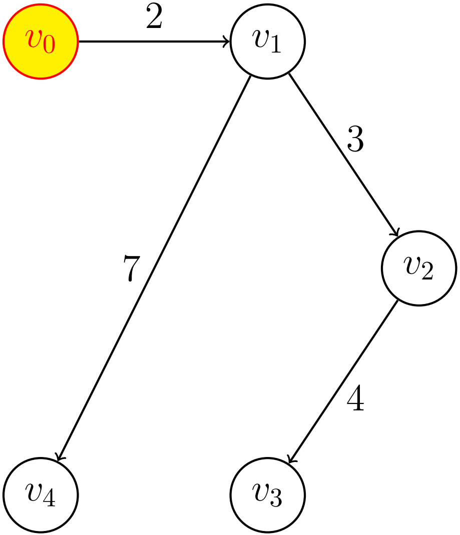


- Initially:
  $S = \{v_0\}, d[v_0] = 0,$
  $d[v_i]$ is $2, +\infty, +\infty, 10,$
  $i = 1, 2, 3, 4;$
- At the first iteration:
  $w = v_1$ is selected, since
  $d[v_1] = 2$ is minimum;
- Then:
  $d[v_2] = \min\{+\infty, 2 + 3\} = 5$
  $d[v_4] = \min\{10, 2 + 7\} = 9$

Transition of variables

Iter.	S	w	$d[w]$	$d[v_1]$	$d[v_2]$	$d[v_3]$	$d[v_4]$
Init.	$\{v_0\}$	$-$	$-$	2	$+\infty$	$+\infty$	10
1	$\{v_0, v_1\}$	v_1	2	2	5	$+\infty$	9
2	$\{v_0, v_1, v_2\}$	v_2	5	2	5	9	9
3	$\{v_0, v_1, v_2, v_3\}$	v_3	9	2	5	9	9
4	All	v_4	9	2	5	9	9

Single-source shortest path of "Example"



correctness of Dijkstra's algorithm

We now prove the correctness of Dijkstra's algorithm by induction on the size of S.

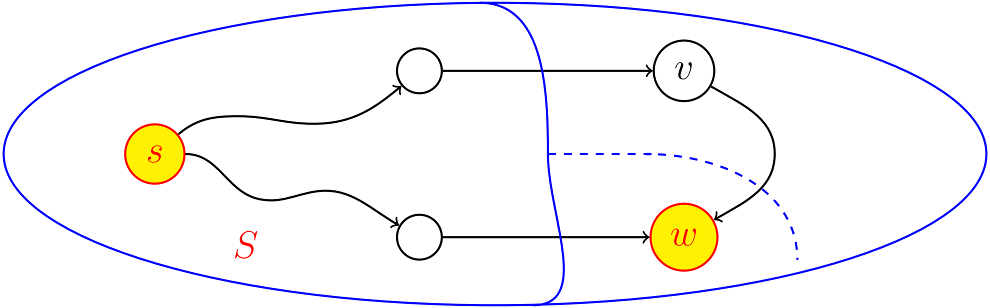
- Basis. $|S| = 1$. The shortest path from s to itself has length 0 and a path from s to v, wholly within S except for v, consists of the single edge (s, v). Thus, $d[v]$ was correctly computed.
  (for $(v \in V - S)$ $d[v] = D[s, v]$;)

correctness of Dijkstra's algorithm(contd)

- Induction Hypothesis. Assume the following statements are true for $|S| = k \geq 1$: S is the partial solution set such that the shortest paths from s to each vertex in S lies wholly in S. For each vertex $v \in V - S$, $d[v]$ is the distance of current shortest path from s to v passing only through vertices of S.

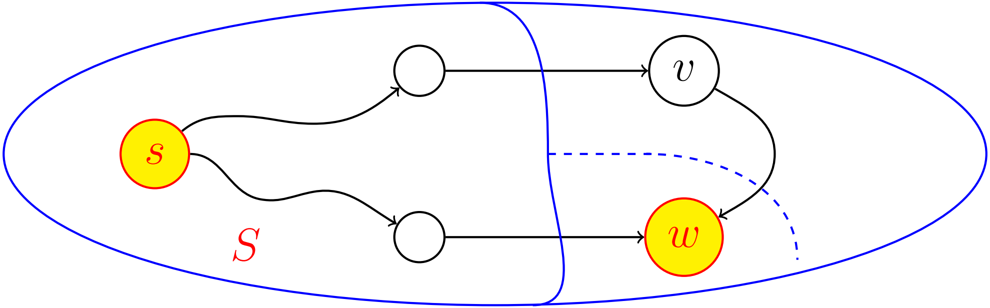
correctness of Dijkstra's algorithm(cntd2)

- Inductive Step. For $|S| = k$, assume $w \in V - S$ such that $d[w]$ is a minimum is chosen and added to S ($|S|$ becomes $k+1$). If $d[w]$ is not the distance of a shortest path from s to w, then there must be a shorter path P such that P contains some vertex other than w which is not in S. Let v be the first such vertex on P.



correctness of Dijkstra's algorithm(cntd3)

- But then the distance from s to v is shorter than $d[w]$, and moreover, the shortest path from s to v lies wholly within S, except for v itself. Thus, by the inductive hypothesis, $d[v] < d[w]$, a contradiction.

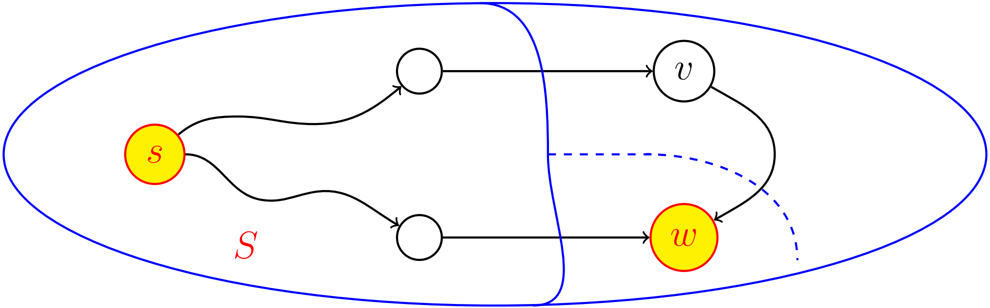


correctness of Dijkstra's algorithm(cntd4)

- We conclude that the path P does not exist and $d[w]$ is the distance of the shortest path from s to w that lies wholly in S. From the update operation

$$d[v] = \min\{d[v], d[w] + D[w, v]\},$$

we know $d[v]$ are correctly computed.



An Efficient Implementation of Dijkstra's Algorithm

- If the graph is represented by an adjacent (distance) matrix, the time complexity of Dijkstra's algorithm is $O(|V|^2)$.
- Dijkstra's algorithm can be made more efficiently by maintaining the graph using adjacency(distance) lists and keeping a priority queue of the nodes not in S. Next slide shows the pseudo code of such an implementation.

Dijkstra's Algorithm(List ver.)

/* s is the source vertex and S is the partial solution set such that the shortest paths from s to each vertex in S lies wholly in S.
For each vertex $v \in V - S$, $d[v]$ contains the distance of current shortest path from s to v passing only through vertices of S.
Here $D[i, j]$ describes distance list*/

Dijkstra's Algorithm(List ver.)2

$S = \{s\}; d[s] = 0;$

for $(v \in V - S)$ $d[v] = D[s, v]$;

for $(v \in V - S)$ construct $d[v]$ into

a minimum heap;

while $(S \neq V)$ {

delete $d[w]$ from the heap and add w to S;

for $(v \in V - S)$ if $((w, v) \in E)$

$\{d[v] = \min\{d[v], d[w] + D[w, v]\}$;

restore the heap condition;}

}

An Efficient Implementation of Dijkstra's Algorithm

- Under this implementation, the time complexity of Dijkstra's algorithm is $O((|V| + |E|) \log |V|)$.

Difference between Minimum Spanning Tree and Shortest Path Spanning Tree

- In Prim's algorithm, vertices and edges are added to a tree one by one, each step choosing the shortest possible edge from V to $V - T$ to add.
- In Dijkstra's algorithm, the edge (u, v) to be added may not be the closest, but minimizing $d(u) + D[u, v]$ within all paths from the source to v.

Shortest Paths in the Graph with Unit Edge Length

- Given a graph $G(V, E)$ whose distances of edges are 1, the shortest path from u to v is the path from u to v with the minimum length (the number of edges). Dijkstra's algorithm may be used to solve this problem.
  However, a simpler algorithm for this problem is BFS algorithm.

All Pairs Shortest Path Problem

- Warshall's algorithm   Floyd's algorithm