Outline

1. Chapter 14: Graphs
   - Shortest Path algorithms continues

Depth First Search

The DFS algorithm moves along one path as far as possible before backtracking and examining other paths off the earlier discovered vertices.

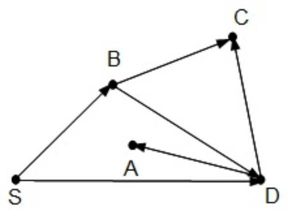
During the DFS execution, each **vertex** goes through three phases:

1. the vertex has not yet been discovered.
2. the vertex has been discovered, but the algorithm has not completed processing of all the vertices accessible from it.
3. we finished processing the vertex and all the vertices reachable from it

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```

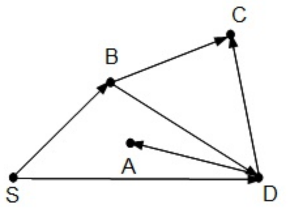


	S	A	B	C	D
par					
st					
et					

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```

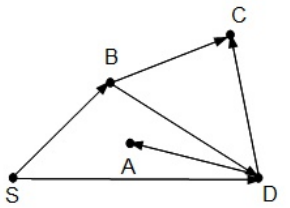


	S	A	B	C	D
par					
st	0	0	0	0	0
et					

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



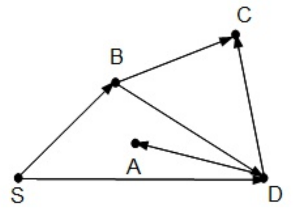
	S	A	B	C	D
par					
st	0	0	0	0	0
et					

t = 0

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S

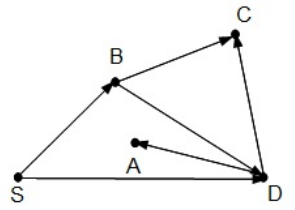
	S	A	B	C	D
par					
st	0	0	0	0	0
et					

t = 0

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S's start time is 0

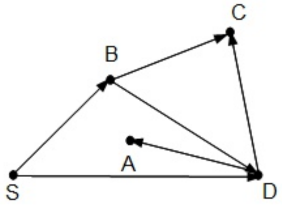
	S	A	B	C	D
par					
st	0	0	0	0	0
et					

t = 0

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S

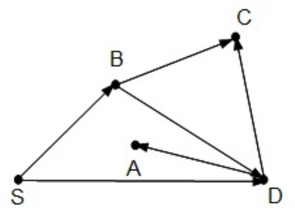
	S	A	B	C	D
par					
st	1	0	0	0	0
et					

t = 1

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S u: B

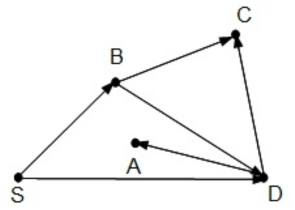
	S	A	B	C	D
par			S		
st	1	0	0	0	0
et					

t = 1

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S   u: B

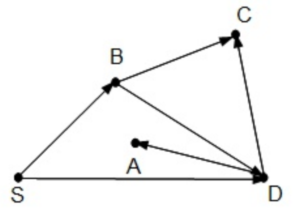
	S	A	B	C	D
par			S		
st	1	0	0	0	0
et					

t = 1

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B

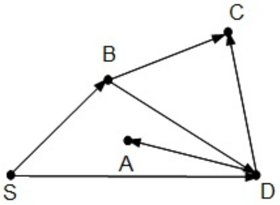
	S	A	B	C	D
par			S		
st	1	0	2	0	0
et					

t = 2

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:C

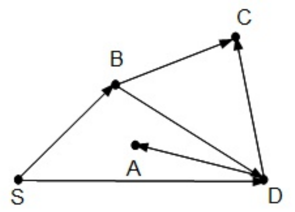
	S	A	B	C	D
par			S	B	
st	1	0	2	0	0
et					

t = 2

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:C

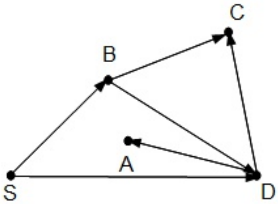
	S	A	B	C	D
par			S	B	
st	1	0	2	0	0
et					

t = 2

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:C

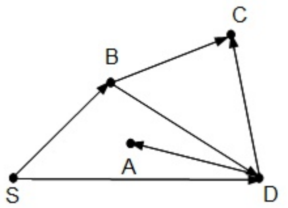
	S	A	B	C	D
par			S	B	
st	1	0	2	3	0
et					

t = 3

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:C

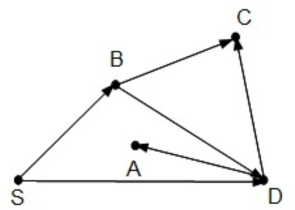
	S	A	B	C	D
par			S	B	
st	1	0	2	3	0
et					

t = 3

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:C

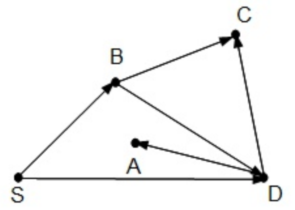
	S	A	B	C	D
par			S	B	
st	1	0	2	3	0
et				4	

t = 4

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:D

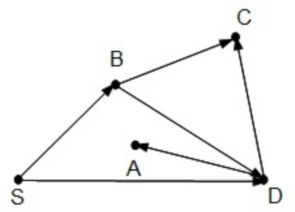
	S	A	B	C	D
par			S	B	B
st	1	0	2	3	0
et				4	

t = 4

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B u:D

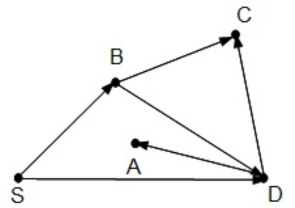
	S	A	B	C	D
par			S	B	B
st	1	0	2	3	0
et				4	

t = 4

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: D

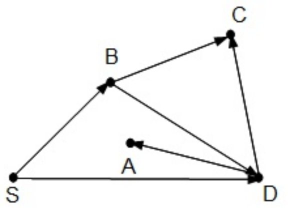
	S	A	B	C	D
par			S	B	B
st	1	0	2	3	5
et				4	

t = 5

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: D u:A

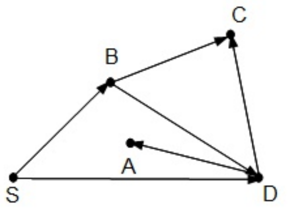
	S	A	B	C	D
par		D	S	B	B
st	1	0	2	3	5
et				4	

t = 5

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: A

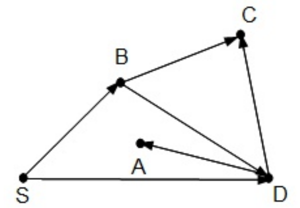
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et				4	

t = 6

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: A

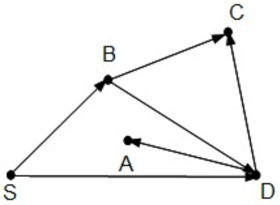
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et				4	

t = 6

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: A

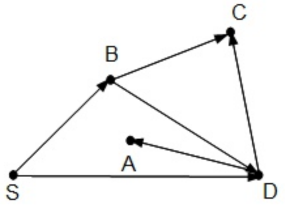
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7		4	

t = 7

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: D u:C

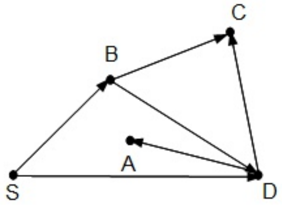
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7		4	

t = 6

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: D u:C

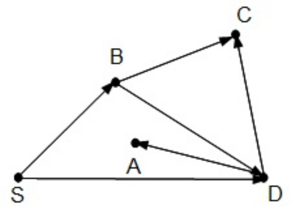
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7		4	8

t = 8

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B

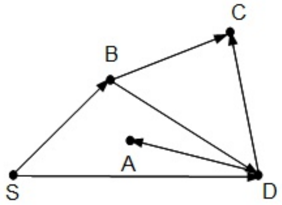
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7		4	8

t = 6

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: B

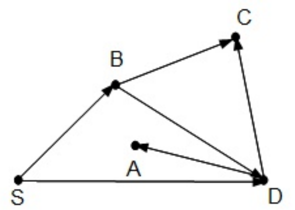
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7	9	4	8

t = 9

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S u:D

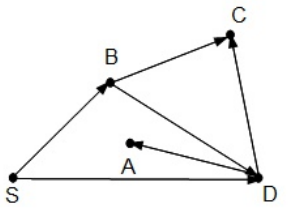
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et		7	9	4	8

t = 9

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: S

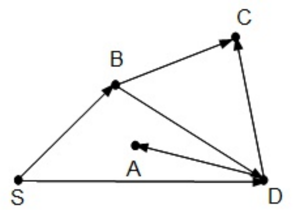
	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et	10	7	9	4	8

t = 10

Depth First Search

```
dfs(g):
   for each vertex v in g:
      set v's start time to 0
   t=0
   for each vertex v in g:
      if v's start time is 0:
         dfs_traverse(g,v)

dfs_traverse(g,v):
   t += 1
   set v's start time to t
   for each vertex u adjacent
to v:
      if u's start time is 0:
         set u's parent to v
         dfs_traverse(g,u)
   t += 1
   set v's end time to t
```



v: A,B,C,D

	S	A	B	C	D
par		D	S	B	B
st	1	6	2	3	5
et	10	7	9	4	8

t = 10

Depth First Search

Running time analysis

`dfs` function processes each vertex in a constant number of times.

`dfs_traverse` processes each edge once and performs a constant number of operations as it processes each edge.

Hence, the overall time is $\Theta(V + E)$.