Theorem 22.9 (white path theorem) The vertex v is a descendant of u in the DFS forest if and only if at time $u.d$, there exists a path from u to v with only white vertices.

Lemma 22.11 A directed graph is acyclic if and only if DFS finds no back edges. Equivalently, a directed graph has cycles if and only if DFS finds back edges.

1 Topological Sort (CLRS §22.4)

Tasks to be performed where some tasks must be completed before others. E.g. prerequisites for courses. Input is

Example

Definition A **topological ordering** of G is an ordering of its vertices $v_1, \cdots, v_n$ so that

Algorithm.
Input:

1. Run

2. Output

Runtime.

Theorem. The given algorithm outputs

Proof.

Remark. In every DAG G, there is a vertex v

2 Strongly Connected Components (SCC) (CLRS §22.5)

Example

Definition A **strongly connected component** (SCC) of a directed graph $G = (V, E)$ is a

The **component graph** G^{SCC} is the graph

Claim. The component graph is a DAG.

Proof.

Consider now DFS on the graph G. We can pretend that this DFS takes place in G^{SCC}, in the following sense: we say that

- a component is discovered when
- a component is finished when
- a component starts

Key lemma: The "pretend" DFS on G^{SCC} induced by the DFS on G is a valid DFS exploration of G^{SCC} .

Proof idea.

Observation. The last vertex to finish

We can now identify the entire left-most component by

SCC(G)

1. Call DFS(G) to compute

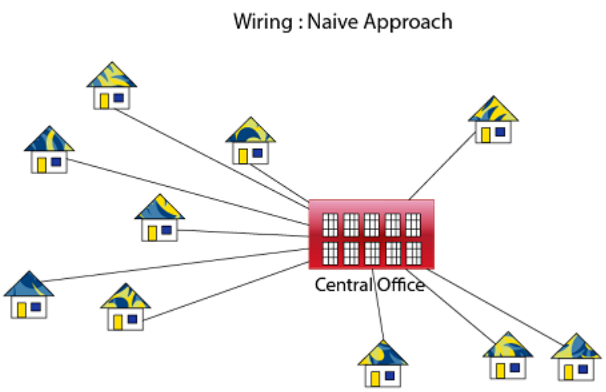
2. Call DFS(G^T) where in the main outer loop consider

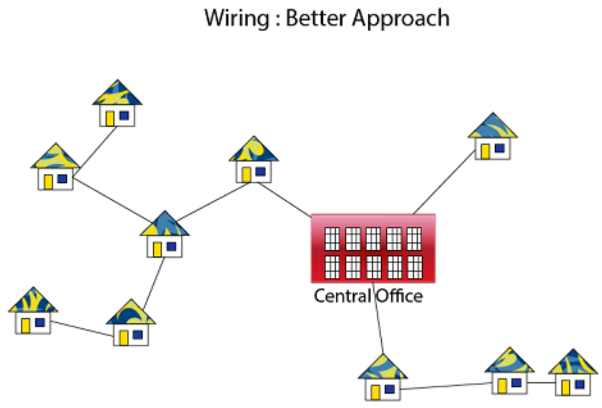
3. Output

Runtime.

Example

3 Minimum Spanning Tree (MST) Problem





3.1 Graph Theory Terminology

Definition A *tree*

Definition A *spanning tree*

Minimum spanning tree problem:

[1] https://www.javatpoint.com/applications-of-minimum-spanning-tree

Example

Why not check all spanning trees?

Example

Remark. A tree on n vertices

$$\leq |E| \leq$$

MST properties. $T \subseteq E$

1. If $|T| = |V| - 1$ and no cycles, then

2. If $|T| = |V| - 1$ and spanning, then

3.2 Greedy choice and generic method (CLRS §23.1)

GenericMST(G, w)

```

while A is not a spanning tree
    find
    A =
return A
```

Definition

- A **cut** of $G = (V, E)$ is
- Edge (u, v) **crosses** a cut
- A cut **respects** A if
- (u, v) is a **light edge** crossing a cut if

Theorem. Let $A \subseteq E$ be included in some MST, $(S, V \setminus S)$ be a cut respecting A, and (u, v) a light edge crossing $(S, V \setminus S)$. Then

Proof.

4 Prim's Algorithm (CLRS §23.2)

4.1 Algorithm Outline and Example

1. Start with
2. Choose an arbitrary
3. Start growing
   (a) Find all
   (b) Choose the
   (c) Add the

Example

4.2 Algorithm Pseudocode & Correctness

MST-PRIM(G, w, r)

```
for u in V
    u.key =
    u.parent =
r.key =
Q =
while Q
    u =
    for v in
        if v in Q and
            v.parent =
            v.key =
```

Correctness

4.3 Time complexity of Prim's

Example

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

Vertex	A	B	C	D	E	F	G	H
Key Value								
Parent								

5 Kruskal's algorithm (CLRS §23.2)

5.1 Plan & Correctness

1. Order edges by
2. T

Correctness.

5.2 Pseudocode

MST-PRIM(G, w, r)

```
A =
for v in
    MAKESET(V)
Sort E in
for (u, v) in
    if
        A =
        UNION
Return
```

5.3 Time complexity of Kruskal's

5.4 Disjoint Forest / Union-Find Data Structure (§21.3)

- parent pointers
- size

1. For any x, find
2. Given C_1, C_2 with roots

Claim. If `x.parent = y`, then

Example

```
initialize v.parent = v, v.size = 1
while x.parent
     x = x.parent
while y.parent
      y = y.parent
if x not equal to y
     if x.size <= y.size
          x.parent =
          y.size =
     else
          y.parent =
          x.size =
```