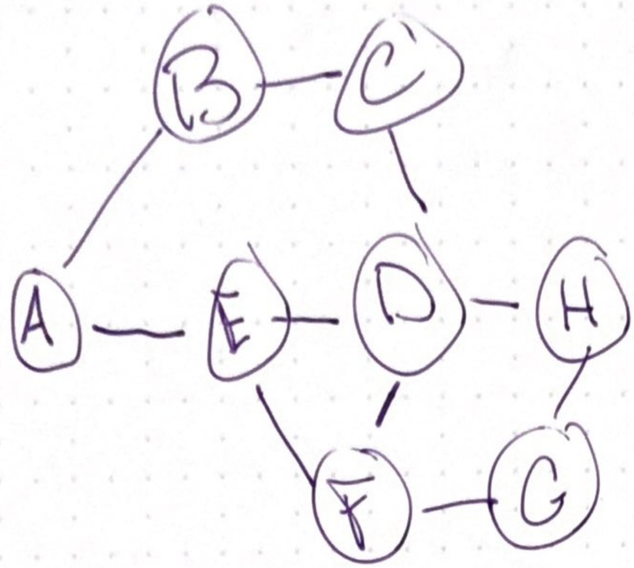
BFT & Recursive



	# length	Prev
A	0	
B	1	A
C	2	B
D	3	C
E		
F		
G		
H		

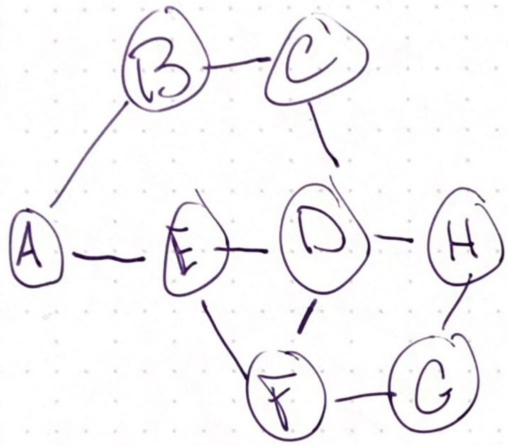

```
Int
del_link(from, to):
{
    t_link *Tmp;
    Tmp = From->to_link;
    while (Tmp)
    {
        if(tmp->End == to)
        {
            tmp->prev = tmp->Next;
            tmp->next->Prev = tmp->prev;
            free (tmp);
            return (1);
        }
        tmp = tmp->next;
    }
    return (0);
```

```
t_Node
{
    char *Name
    t_Node *Prev
    t_Link *to_link
}

t_link
{
    t_link *Next
    t_Node *End
    t_link *prev
}
```

C++ 2 Recursive

t_link Add ->
& weight! (logic)



	# length	Prev
A	0	
B	1	A
C	2	B
D	3	C
E		
F		
G		
H		


```
Expand_shortest(t_Node *End)
{
  t_Node *tmp;
  tmp = *End;
  While (tmp)
  {
   del_link (tmp->to_link, prev->to_link)
   invert_weight (prev->to_link, tmp);

  }
```

```
invert_weight(t_link *P_link,
              t_Node *Dst_Node)
{
  t_link *tmp;
  tmp = p_link;
  While (tmp)
  {
  If (tmp->End == Dst_Node)
  {
     tmp->weight = -weight
     return (1);
  }
  tmp = tmp->next;
}
```