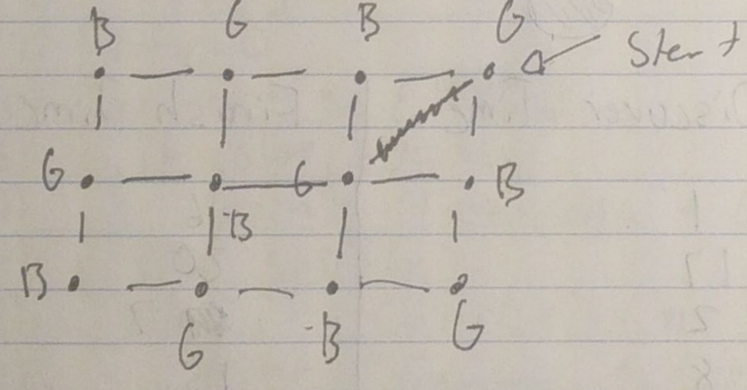
HW # 6

1) The worst case run time for BFS ~~search~~ on an adjacency matrix is: $\boxed{O(V^2)}$

2) Assume graph G has n vertices and that for one vertex V its neighbors will be considered rivals.

Example



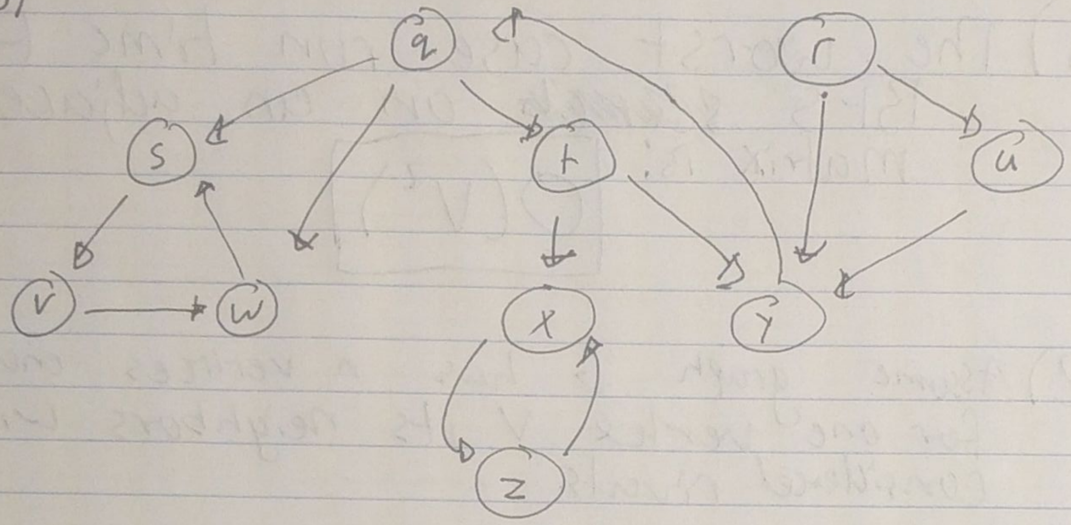
If a BFS is performed on a graph such as this it will have a time complexity: which is desired.

$$O(n+r)$$

If a wrestler has already ~~des~~ been designated do not designate it

If it is a rival designate neighbors as non rivals and vice versa.

3)



Vertices	Discover Time	Finish Time
q	1	16
r	17	20
s	2	7
t	8	15
u	18	19
v	3	6
w	4	5
x	9	12
y	13	14
z	10	11

Tree: (q,s), (s,v), (v,w), (q,t), (t,x), (x,z), (t,y), (r,u)
Forward: (q,w)
Back: (w,s), (z,x), (y,q)
Cross: (r,y), (u,y)

4)

	White		Gray		Black	
White	Tree	Back	Back	Cross	Cross	
	Forward	Cross				
Gray	Tree	Forward	Tree	Forward	Tree	Forward
			Back		Cross	
Black			Back		Tree	Forward
					Back	Cross

5) DFS

```
for all x ε V
    color[x]=w
    p[x]=nil
time=0
for each ~~x~~ ε V
    if color[x]=w
        print "Tree edge" + coordinate
    ~~else~~ visit(x)
    else if color[x] = b
        print "Forward" + coordinate
    else if d[x] > d[y]
        print "Cross" + coordinate
    else
        print "back" + coordinate
color[x]=5
f[x] =(++ time)
```

6) DFS

```
time=0
k=1
for all x ∈ V
    color[x]=w
    P[x]=nil
for all x ∈ V
      if color[x]==w
      cc[x]=k
      k++
      visit (x)
```