CS & IT ENGINEERING

Operating Systems

Process Synchronization/ Co-ordination

Lecture No. 12





By- Dr. Khaleel Khan sir



TOPICS TO BE COVERED

Parbegin-Parend

Fork - Join

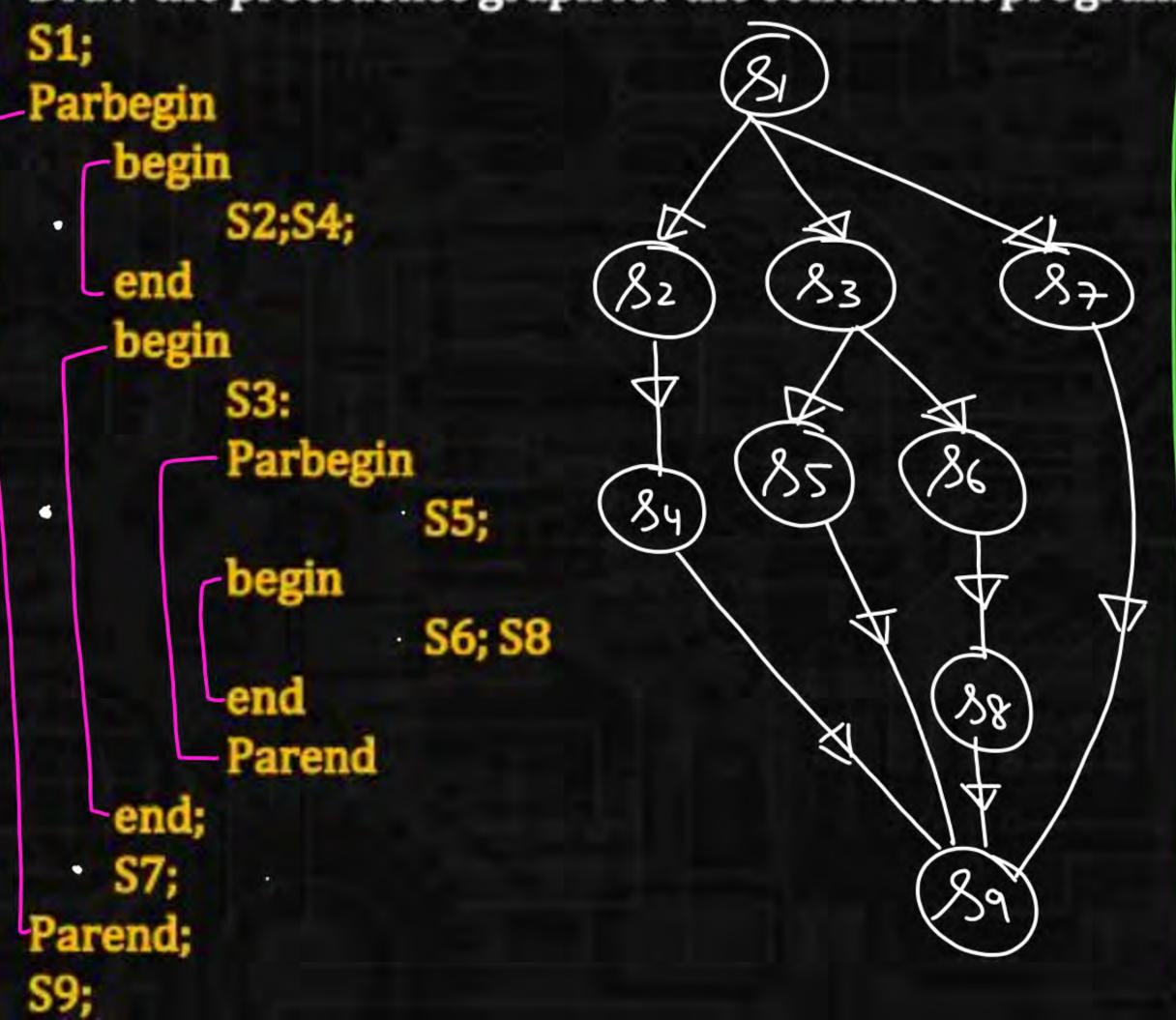
Problem Solving

Partegin-Parend Non-Implementable with Parbegin-Parend alone Such tyraphs are Implementable Burbegm hith Semaphores; BSEM a, b, <, d, e, f, g = {Ø} Cobegin than begin S1; V(a); V(b); end Parend begin P(a); 82; 84; V(c); V(d); enf Parhegin begin P(b); 83; V(e); end begin P(c); 85; V(f); end begin P(d); P(e); 86; V(g); end begin P(f); P(g); 87; end ('send

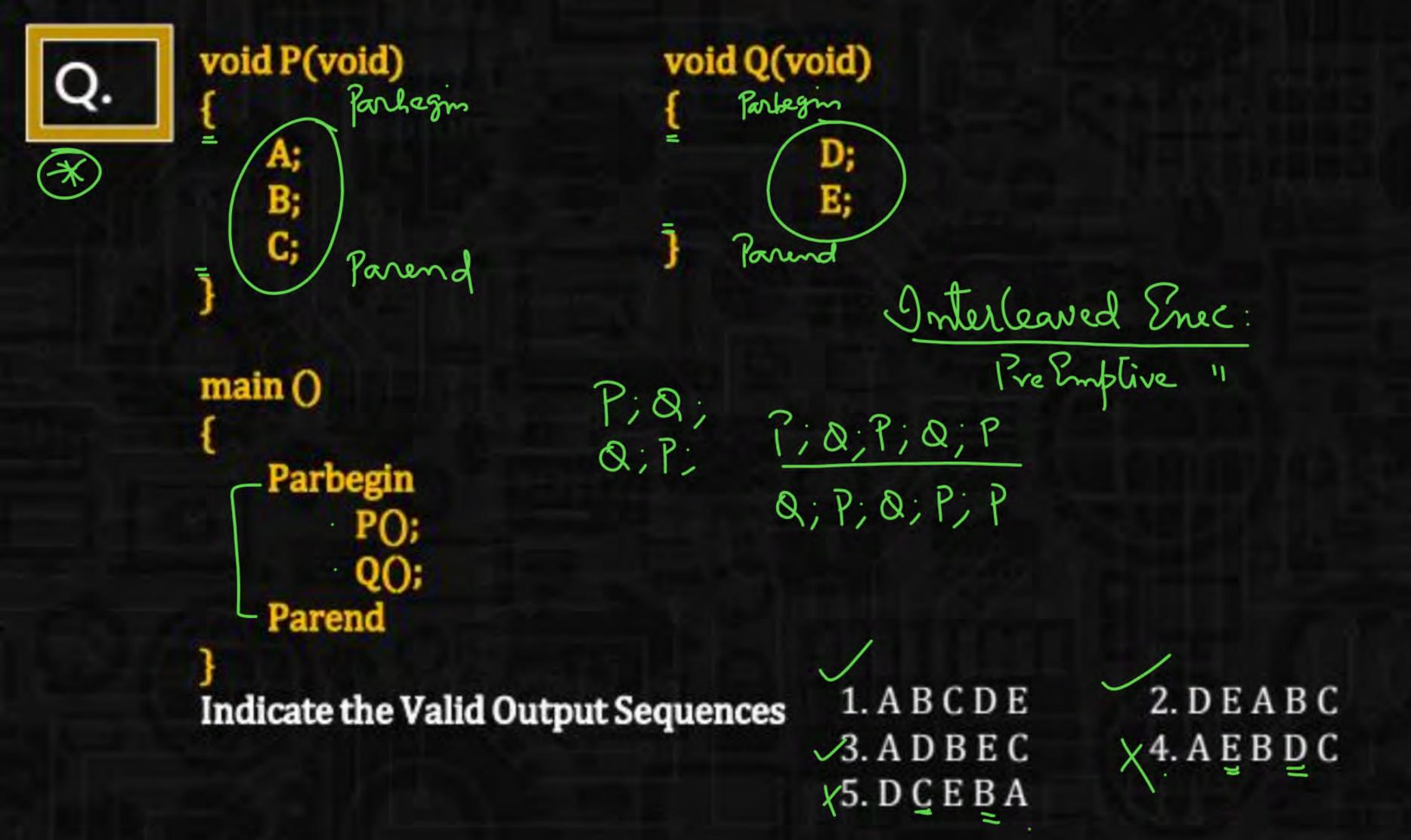
Q.

Draw the precedence graph for the concurrent program given below:





Peterson Soh





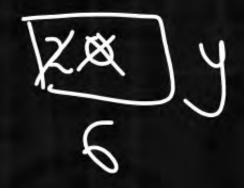
Q.

Coend

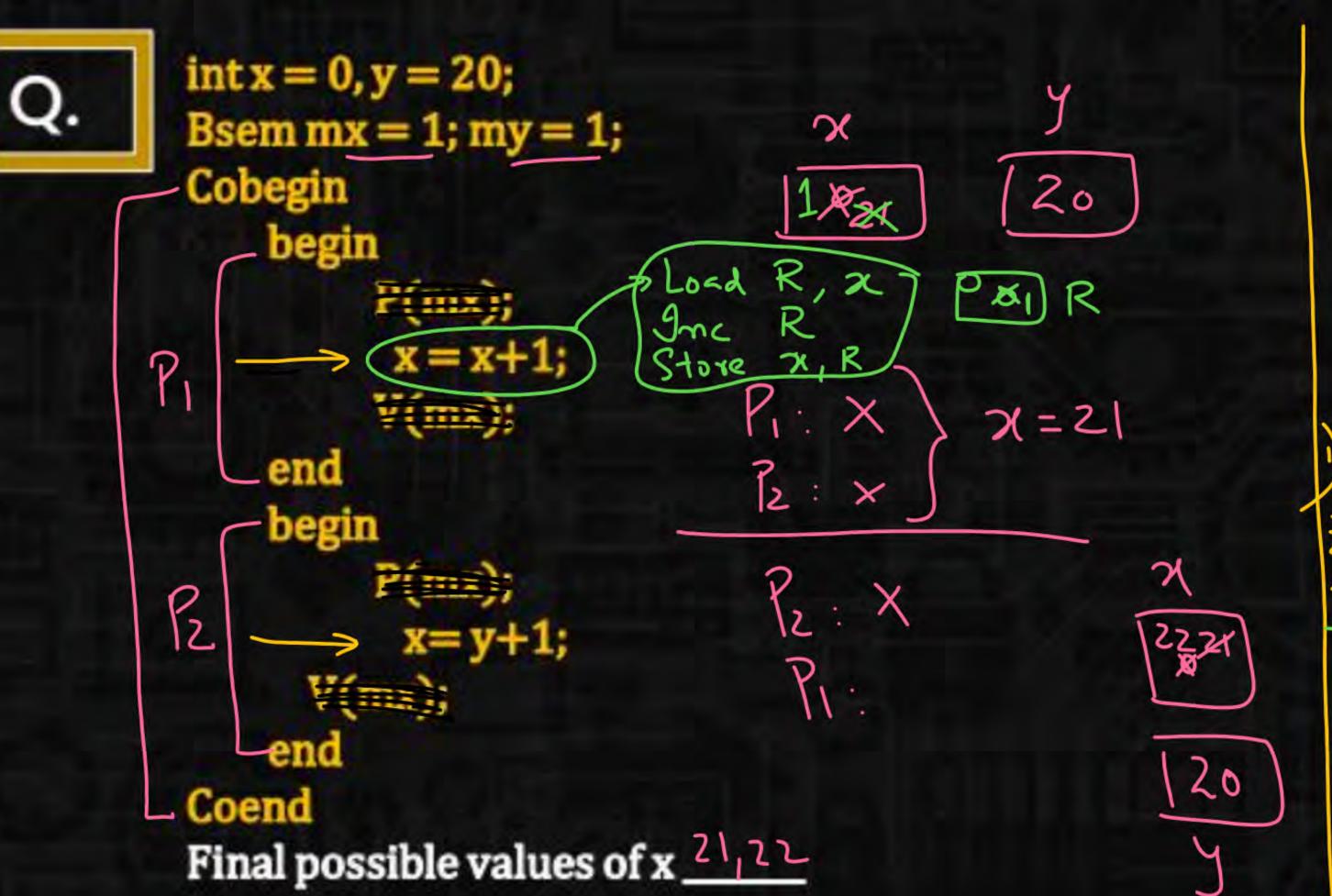
Final values of x & y

I)
$$x = 1$$
; $y = 2$
II) $x = 1$; $y = 3$
III) $x = 4$; $y = 6$



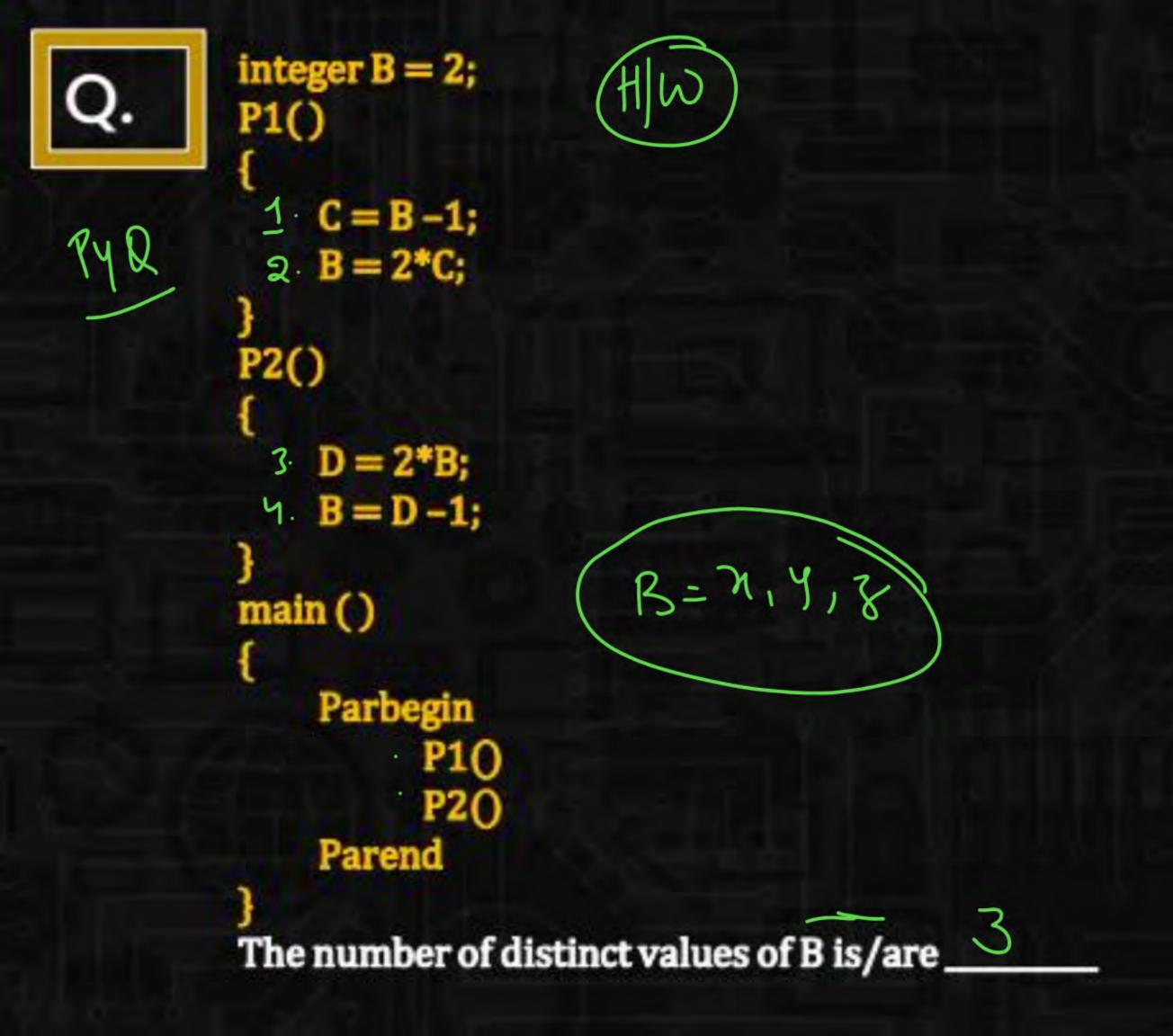






Values 9 X = Pr. L; In; Pre

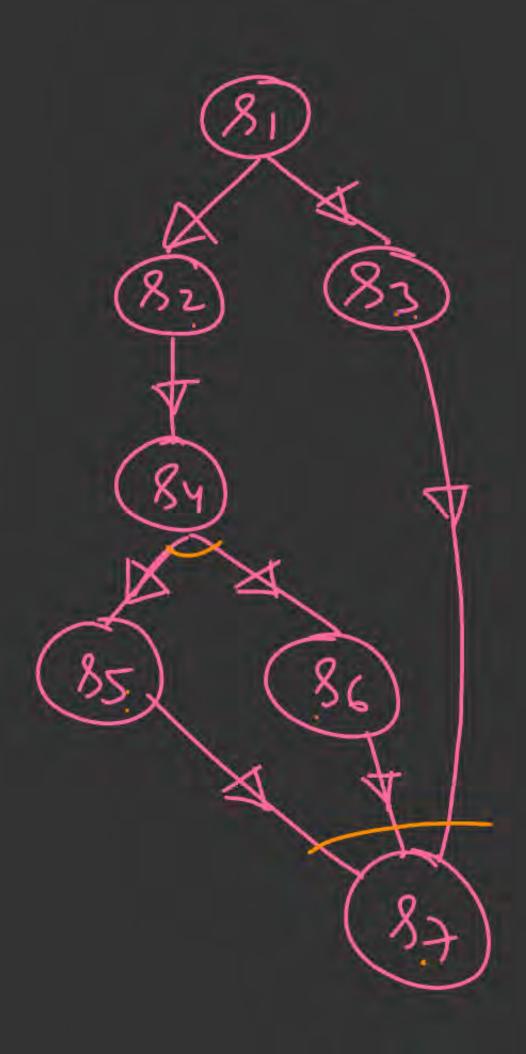
Pr. Store



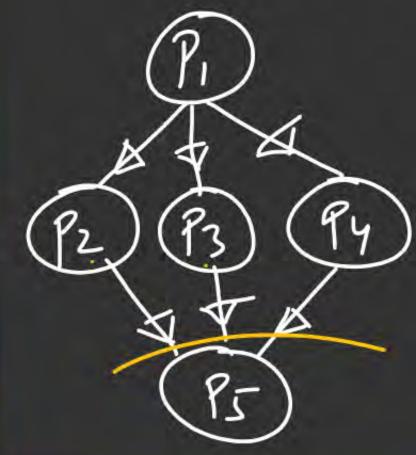


```
Test: Pi
                                                         Test: PZ
                                                0
int count = 0;
                                             Count
void test()
                                 1=1,5
   int i, n = 5;
                                                        count ++
                                  Count ++
   for(i = 1; i \le n; ++i)
   count = count+1;
           Ly NOT Atomic
main ()
  Parbegin
                                          Man (count): 10
      - test();
      - test();
   Parend
                                                    Mim + 5
What is the minimum and maximum value of count?
                                                     Mim +
```

fook - Join integer Count = 2; Syntam q Josk Li Execution of Josh results in starting 2-Computation Concurrently X: Jin (Gount) Jim (int Count) I. Which is Immediately after fook . Count = Count - 1. II. which is at latel L' 2. y (count to) them



```
Count = 3;
       fook X
        BS;
goto Z;
   し: 83; 956 元;
X: 186;
Z: John (Count),
87;
```

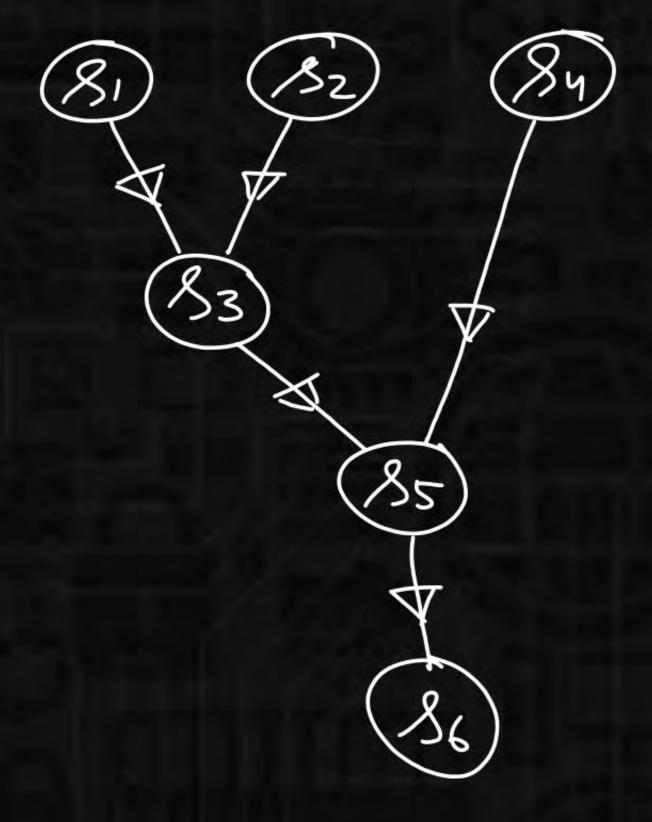


Z: John (Count).
PS: Count = 3; L: Pork X

```
fork L3;
  fork L4;
  S1;
       L1: Join N
        L2: Join M
goto L1;
       L4: S4;
  goto L2;
  next: S6;
```

Draw the Precedence Graph





```
fork L3;
fork L4;
      L1: Join N
 goto L1;
goto L2;
 next: S6;
```

Draw the Precedence Graph

