1. **Safe States**
   1. Snapshot 1 – 12 resource instances total
      1. Safe –– <P2, P1, P3, P4>
   2. Snapshot 2 – 12 resource instances total
      1. Unsafe –– Starting with any process will cause deadlock, there is only 1 resource left after the allocation and all of the processes require 2 or more resources to finish
   3. Snapshot 3 – 12 resource instances total
      1. Unsafe –– Starting with any process will cause deadlock, there is only 1 resource left after the allocation and all of the processes require 2 or more resources to finish
2. **“wait-for” condition**
   1. Pros—
      1. Every process that can successfully run will have all of the resources it needs to run. A process will never get stopped in the middle of running and will always run completely.
   2. Cons—
      1. If there are not enough available allocated resources no process will be able to finish, such as in Snapshots 2 and 3 from the first question. There are not enough extra resources after what has already been allocated, therefore *nothing* will be able to finish. There is no way to un-allocate resources.
3. **3 Processes 4 Resources**
   1. Deadlock will not occur because even if each process has one of the resources there will still be 1 “free” resource. This means that there is guaranteed to be at least one process that can have a full 2 resources.
   2. m+(n-1) resources will be required to prevent deadlock.
   3. ­n=r/m the maximum number of resources a single process can request is the total number of resources divided by the total number of processes.