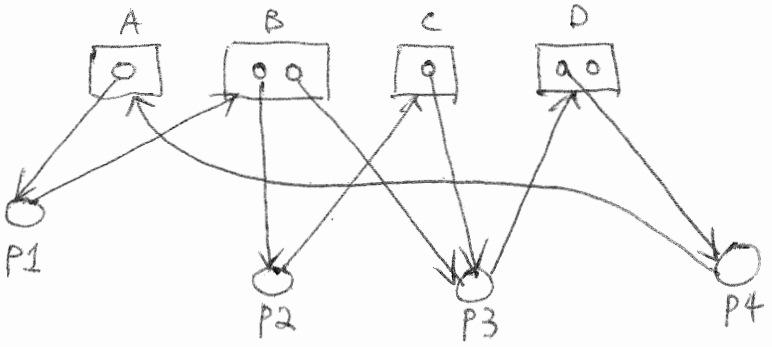
Draw a resource allocation graph to represent the state of the following system, and then answer questions.

The system has 4 resource types and 4 processes. Resource types A and C each has 1 instance, while resource types B and D each has 2 instances. Currently, Process 1 holds 1 instance of A and requests for 1 instance of B; Process 2 holds 1 instance of B and requests for 1 instance of C; Process 3 holds 1 instance of B and 1 instance of C, and requests for 1 instance of D; Process 4 holds 1 instance of D and requests for 1 instance of A.

Draw the resource allocation graph in the following:



Question: Is the system currented deadlocked? If so, list the set of processes that are in a deadlock; if not, show how all the processes can complete (assuming they will not raise new requests).

NO.

P3 → P2 → P1 → P4

P3 → P1 → P2 → P4

P3 → P1 → P4 → P2

5/10

5/10

(giving one sequence is good enough)

4. Apply the banker's algorithm to answer the following question. Let the following matrixes represent the current state of a system, where there are 3 resource types (A, B and C) and 3 processes (P1, P2 and P3).

The MAX matrix (i.e., maximum needs of resource claimed by the processes)

	A	B	C
P1	1	2	2
P2	1	2	1
P3	1	1	1

The Allocation matrix (i.e., current allocation of resource instances to the processes)

	A	B	C
P1	1	2	0
P2	0	1	1
P3	1	0	1

The Avail vector (i.e., the number of resource instances that are currently available)

A	B	C
0	1	1

Question: If process P1 makes a request (0 0 1), can the request be immediately granted? [5pts] Explain why based on the banker's algorithm. [10pts]

Need

	A	B	C
P1	0	0	2
P2	1	1	0
P3	0	1	0

2/10 ① $Req_1 < Need_1$ (0 0 1) < (0 0 2) valid ✓

2/10 ② Avail (0 0 1) < (0 1 1) ~~as~~ possible ✓

6/10 ③ pretend granting

Avail		
A	B	C
~~1~~	~~2~~	~~1~~
0	1	0

Alloc		
P1	1	2	1
P2	0	1	1
P3	1	0	1

Need		
0	0	1
1	1	0
0	1	0

seq: P3 → P1 → P2
↳ P2 → P1

safe!

Yes ← 5/5