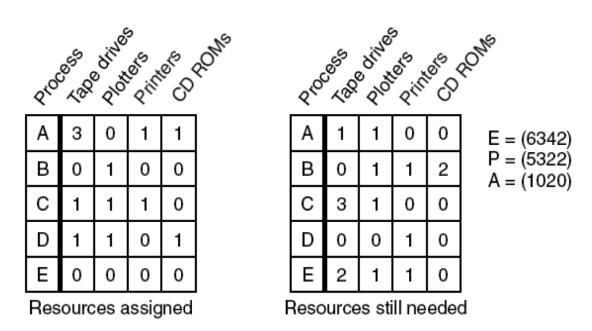
Banker's Algorithm

The Banker's algorithm is run by the operating system whenever a process requests resources. The algorithm **avoids** deadlock by denying or postponing the request if it determines that accepting the request could put the system in an unsafe state (one where deadlock could occur). When a new process enters a system, it must declare the maximum number of instances of each resource type that it may ever claim; clearly, that number may not exceed the total number of resources in the system. Also, when a process gets all its requested resources it must return them in a finite amount of time.



Algorithm for checking to see if a state is safe:

- 1. Look for row, R, whose unmet resource needs all \leq A. If no such row exists, system will eventually deadlock since no process can run to completion
- 2. Assume process of row chosen requests all resources it needs and finishes. Mark process as terminated, add all its resources to the A vector.
- 3. Repeat steps 1 and 2 until either all processes marked terminated (initial state was safe) or no process left whose resource needs can be met (there is a deadlock).

```
IN PUT: // take input from file
4 // no. of processes
5 // no. of resources
11213 // Max. resource Matrix
22210
21310
11221
10211 // Resource allocation Matrix
20110
11010
11110
00212 // resource available
OUT PUT:
Need Matrix:
0\ 1\ 0\ 0\ 2
0\ 2\ 1\ 0\ 0
10300
00111
Safe sequence is:
DACB
Change in available resource matrix:
11322
21533
32543
52653
```