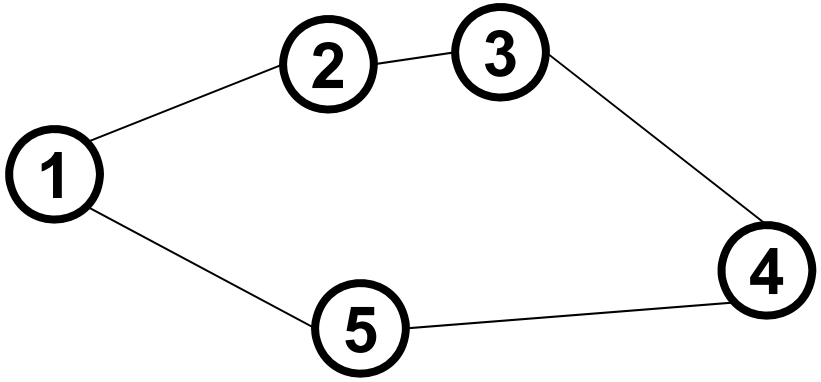
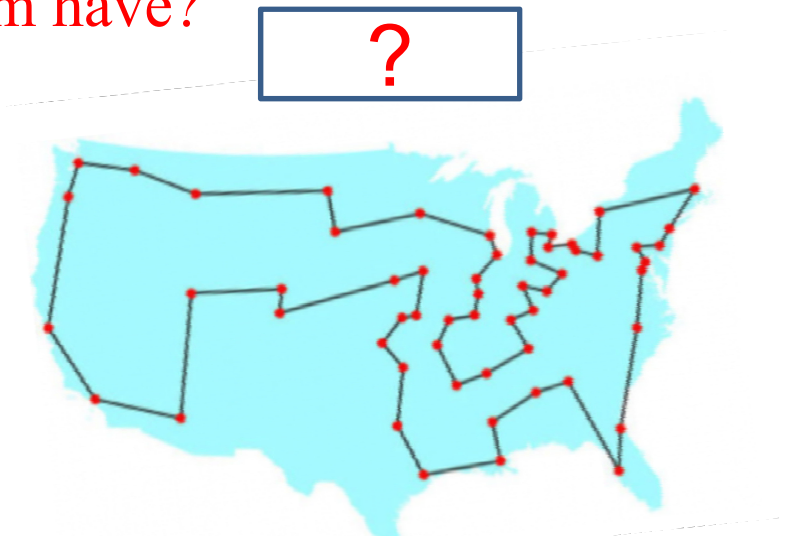
1st Week Lab Session

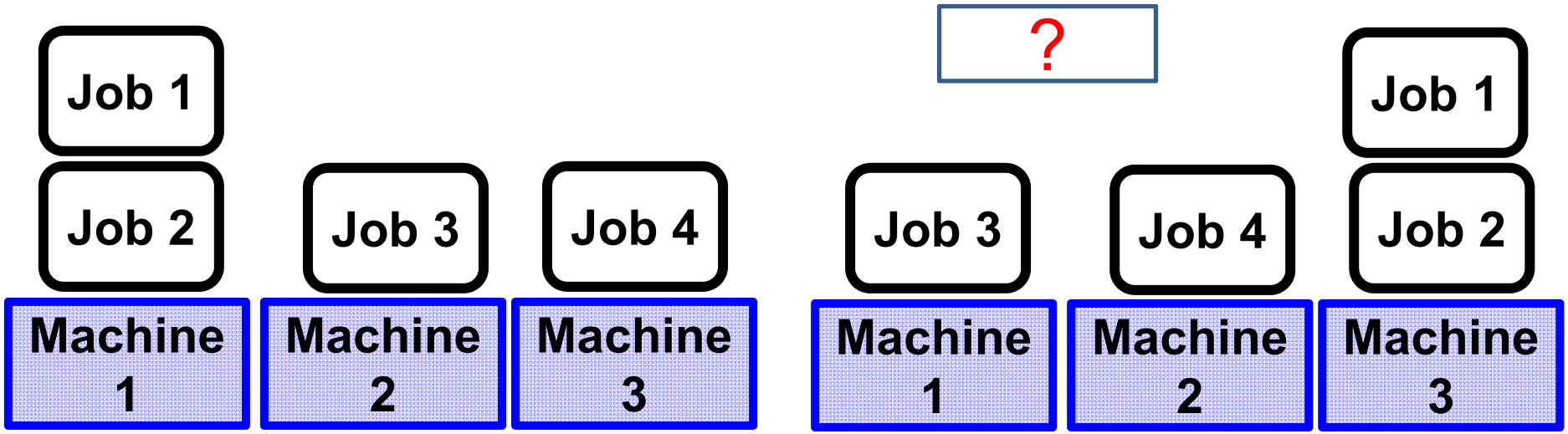
Prepare a PowerPoint file to explain how to solve each question. You will be asked to explain how to solve each question using your file at the end of lab session (e.g., 20:20 - 20:50)

Question 1: An *n*-city TSP problem (i.e., to find the shortest tour to visit all the given *n* cities and return to the start city). The problem size is the total number of different tours. This is not *n*! since many tours can be viewed as the same tour (with respect to the tour length). For example, in the following 5-city TSP problem, the tour 123451 is the same as 154321, 234512, 215432 and some other tours. How many different tours does an *n*-city TSP problem have?



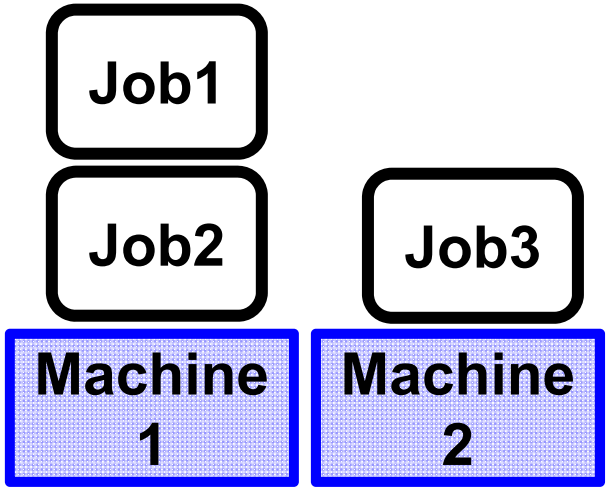


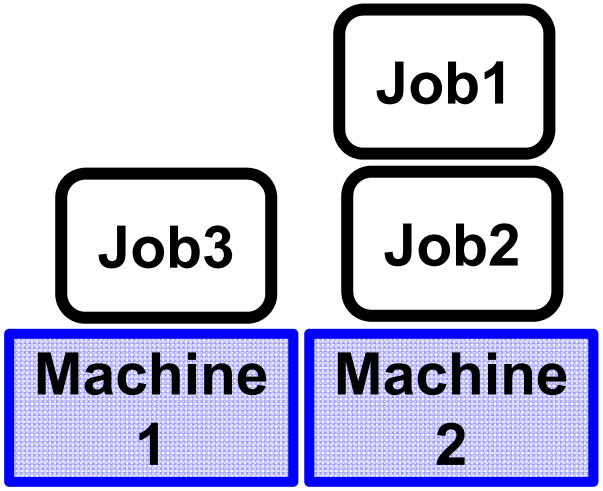
Question 2: A 3-machine 4-job load balancing problem. This problem is to find the best assignment of four jobs to three identical machines. Note that some different assignments are viewed as the same solution since all machines are identical. For example, the assignment of all jobs to Machine 1 is viewed as the same solution as the assignment of all jobs to Machine 2 (or Machine 3). The following two assignments are viewed as the same solution. How many different solutions does the 3-machine 4-job load balancing problem have?



This problem is the same as a grouping problem of four jobs into three groups: {{J1, J2}, {J3}, {J4}} is the same as {{J3}, {J4}, {J1, J2}}.

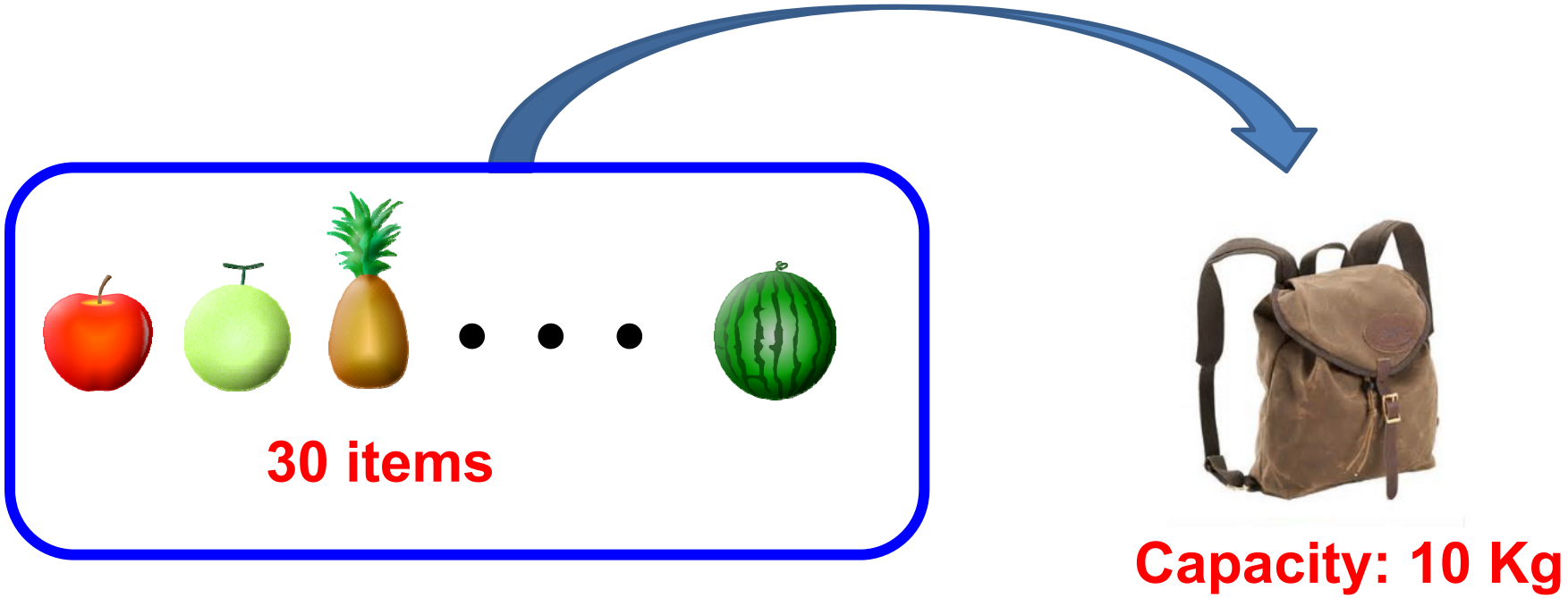
Question 3: A 2-machine *n*-job load balancing problem. As in Example 2, some different assignments (e.g., two assignments below) are viewed as the same solution since the two machines are identical. How many different solutions does the 2-machine *n*-job load balancing problem have? ?





A 2-machine *n*-job load balancing problem is different from subset selection from *n* jobs. In the above two figures, the selection of {Job 1 and Job 2} is different from the selection of {Job 3} in subset selection (whereas these two solutions are the same in load balancing).

Question 4: A 30-item knapsack problem (to find the best item set of the given 30 items under a capacity constraint). The set of all different solutions includes the selection of no items and the selection of all items. How many different solutions does the 30-item knapsack problem have? ?



This problem is the same as subset selection.

Question 5: How many items are needed in an *m*-item knapsack problem to have a similar search space size to a 1000-city TSP problem.

Question 6: How many cities are needed in an *n*-city TSP problem to have a similar search space size to a 1000-item knapsack problem.