Q1

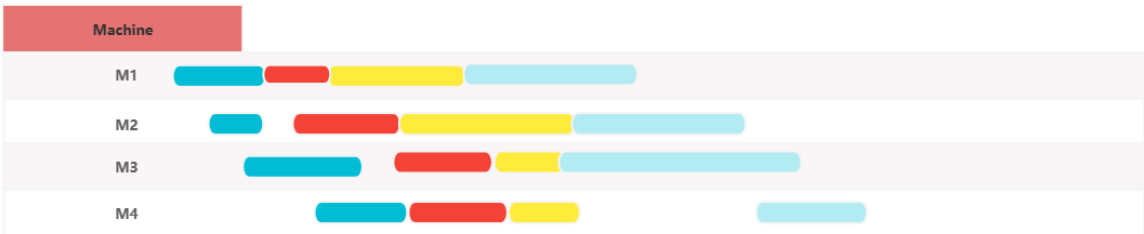
depth1						
1	2{5.38}	3{7.30}	4{8.69}	5{7.10}	6{4.54}	7{6.48}
2	1{8.48}	3{7.30}	4{10.48}	5{8.54}	6{5.10}	7{6.68}
3	1{10.48}	2{7.38}	4{12.39}	5{10.38}	6{6.54}	7{7.30}
4	1{8.07}	2{6.76}	3{8.59}	5{7.32}	6{5.30}	7{7.30}
5	1{7.72}	2{6.07}	3{7.83}	4{8.56}	6{4.54}	7{6.55}
6	1{9.01}	2{6.48}	3{7.83}	4{10.34}	5{8.39}	7{6.41}
7	1{10.96}	2{8.06}	3{8.59}	4{12.39}	5{10.40}	6{6.41}

By using the greedy solution of tour length 10, there are 8 depth 2 nodes can be terminated.

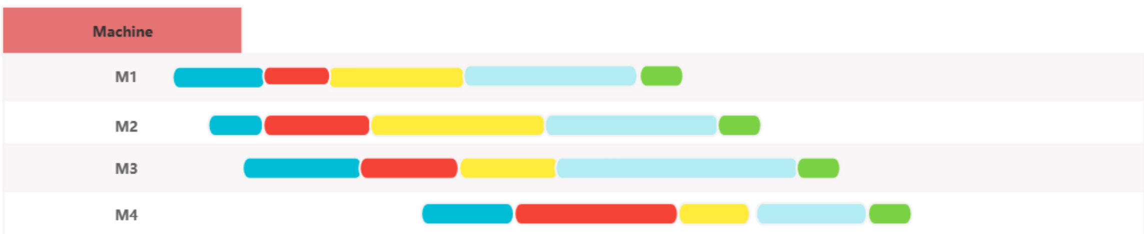
They are 0-2-4, 0-3-1, 0-3-4, 0-3-5, 0-6-4, 0-7-1, 0-7-4, 0-7-5.

Q2

Suppose there are 4 jobs and 4 machines, then select "Blue lattice" as the first job. The schedule is shown in followings:



Then compress the wait time for all tasks except the first for all machines. And add a period which is the shortest time in the remains jobs of the last machine used by "Green lattice" . Then select the maximum total time from all machines as for lower bound.



Q3

For 10-machine *n*-job flowshop scheduling problems, in one hour, the size of the problem that can be solved depends on several factors, including computer performance, algorithm efficiency, and problem characteristics. Here's a simple estimate, assuming the use of a high-performance computer and an efficient branch and bound algorithm for solving.

Let's assume that our algorithm can process 1000 nodes per second (which is a relatively high speed, depending on the optimization level of the algorithm and actual hardware performance). Then, in one hour, the number of nodes the algorithm can process is:

$$1000 \text{ nodes/second} \times 3600 \text{ seconds/hour} = 3,600,000 \text{ nodes/hour}$$

Now we need to consider how much time each node takes to compute. Since we're dealing with a 10-machine flowshop scheduling problem, the number of nodes may exponentially increase with the number of jobs, n. Let's assume each node takes an average of 1 millisecond to compute (which is also a relatively fast estimate). Then, the size of the problem the algorithm can handle in one hour is:

$$3,600,000 \text{ nodes/hour} \times 0.001 \text{ seconds/node} = 3,600 \text{ nodes/hour}$$

This means that for a 10-machine flowshop scheduling problem, assuming each node takes an average of 1 millisecond to compute, the algorithm can roughly handle a problem size of 3600 in one hour. This is a very rough estimate, and the actual situation may vary depending on computer performance, algorithm efficiency, and problem characteristics.