Assignment # 8

1. Consider the simple scheduling problem where we have a set of independent tasks running on a fixed number of processors, and we wish to minimize finishing time.

   How would a <u>list</u> (<u>first fit,</u> <u>no preemption)</u> strategy schedule tasks with the following IDs and execution times onto four processors? Answer using Gantt chart.

(T1,4) (T2,1) (T3,3) (T4,6) (T5,2) (T6,1) (T7,4) (T8,5) (T9,7) (T10,3) (T11,4)

T1	*T1*	*T1*	*T1*	*T8*	*T8*	*T8*	*T8*	*T8*										
T2	*T5*	*T5*	*T6*	*T9*	*T9*	*T9*	*T9*	*T9*	*T9*	*T9*								
T3	*T3*	*T3*	*T7*	*T7*	*T7*	*T7*	*T11*	*T11*	*T11*	*T11*								
T4	*T4*	*T4*	*T4*	*T4*	*T4*	*T10*	*T10*	*T10*										

11 Units

Now show what would happen if the times were sorted non-decreasing.

(T2,1) (T6,1) (T5,2) (T3,3) (T10,3) (T1,4) (T7,4) (T11,4) (T8,5) (T4,6) (T9,7)

T2	*T10*	*T10*	*T10*	*T8*	*T8*	*T8*	*T8*	*T8*										
T6	*T1*	*T1*	*T1*	*T1*	*T4*	*T4*	*T4*	*T4*	*T4*	*T4*								
T5	*T5*	*T7*	*T7*	*T7*	*T7*	*T9*	*T9*	*T9*	*T9*	*T9*	*T9*	*T9*						
T3	*T3*	*T3*	*T11*	*T11*	*T11*	*T11*												

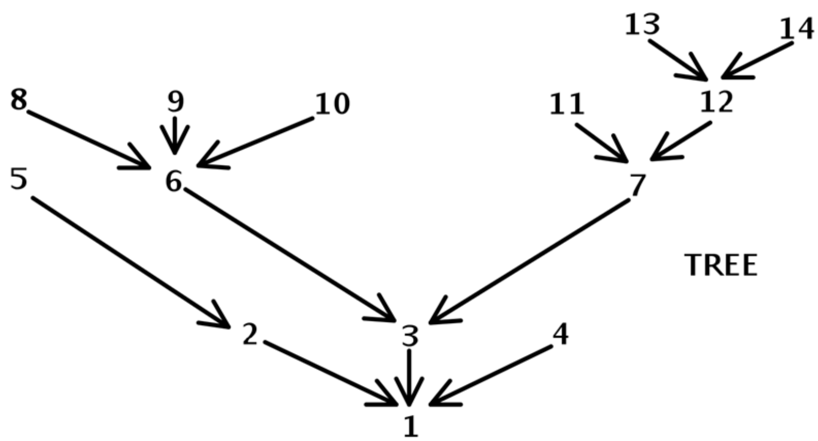
13 Units

Now show what would happen if the times were sorted non-increasing.

(T9,7) (T4,6) (T8,5) (T1,4) (T7,4) (T11,4) (T3,3) (T10,3) (T5,2) (T2,1) (T6,1)

T9	*T9*	*T9*	*T9*	*T9*	*T9*	*T9*	*T10*	*T10*	*T10*									
T4	*T4*	*T4*	*T4*	*T4*	*T4*	*T3*	*T3*	*T3*	*T2*									
T8	*T8*	*T8*	*T8*	*T8*	*T11*	*T11*	*T11*	*T11*	*T6*									
T1	*T1*	*T1*	*T1*	*T7*	*T7*	*T7*	*T7*	*T5*	*T5*									

10 Units



2. Consider adding two additional tasks numbered 15 and 16 that are siblings of 13 and 14. These four tasks must be completed before 12 is started.

a) Write the Gantt chart down that shows the new schedule associated with this enhanced tree

16	*13*	*12*	*7*	*4*	*1*													
15	*11*	*9*	*6*	*3*														
14	*10*	*8*	*5*	*2*														

b) Write down the Gantt chart that is associated with the corresponding anti-tree, in which all arc are turned in the opposite direction. Use the technique of reversing the schedule from (a)

1	*4*	*7*	*12*	*13*	*16*													
	3	*6*	*9*	*11*	*15*													
	2	*5*	*8*	*10*	*14*													

c) Write down the Gantt chart associated with the anti-tree of b), where we now use the priorities obtained by treating lower numbered tasks as higher priority ones

1	*2*	*5*	*8*	*11*	*13*	*16*												
	3	*6*	*9*	*12*	*14*													
	4	*7*	*10*		*15*													

d) Comment on the results seen in (b) versus (c), providing insight as to why they are different and why one is better than the other.

The approach of (a) gives a high priority number to nodes that are far away from the root; moreover, those at the same distance away from the root are ordered so those that are intermediaries on long paths are assigned highest priority numbers. The approach of (c) reverses the semantics of priority numbers, so those with higher numbers are given lower priorities. This essentially ignores the critical path that ends in 13, 14, 15 and 16, giving preference to 8, 9, 10 and 11 over 12 which stands in the way of those at the leaves of the most critical path. In contrast, (b), by reversing the schedule from (a), takes advantage of the information already encapsulated in its rankings.