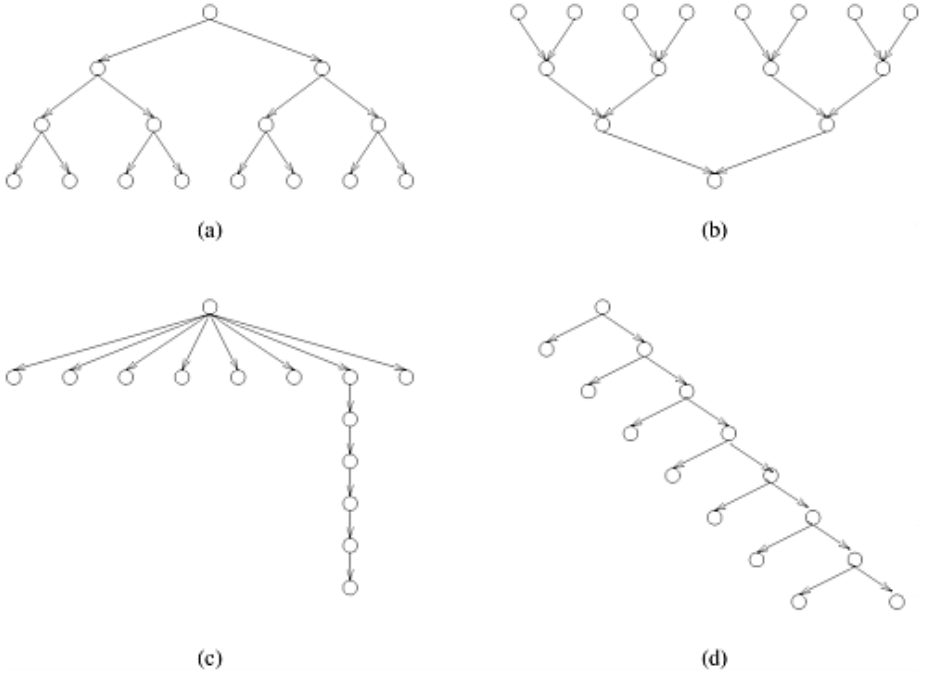
Assignment 1. *For the task graphs given in Figure 3.42, determine the following:*

1. *Maximum degree of concurrency.*
2. *Critical path length.*
3. *Maximum achievable speedup over one process assuming that an arbitrarily large number of processes is available.*
4. *The minimum number of processes needed to obtain the maximum possible speedup.*
5. *The maximum achievable speedup if the number of processes is limited to (a) 2, (b) 4, and (c) 8.*



Assignment 2. *Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming that it does not take any time to assign work to a process, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.*

Assignment 3. *Consider the all-reduce operation in which each processor starts with an array of m words, and needs to get the global sum of the respective words in the array at each processor. This operation can be implemented on a ring using one of the following three alternatives:*

i. *All-to-all broadcast of all the arrays followed by a local computation of the sum of the respective elements of the array.*

ii. *Single node accumulation of the elements of the array, followed by a one-to-all broadcast of the result array.*

iii. *An algorithm that uses the pattern of the all-to-all broadcast, but simply adds numbers rather than concatenating messages.*

1. *For each of the above cases, compute the run time in terms of m, t_s, and t_w.*

2. *Assume that $t_s = 100$, $t_w = 1$, and m is very large. Which of the three alternatives (among (i), (ii) or (iii)) is better?*

3. *Assume that $t_s = 100$, $t_w = 1$, and m is very small (say 1). Which of the three alternatives (among (i), (ii) or (iii)) is better?*

Assignment 4. *(Amdahl's law) If a problem of size W has a serial component W_S, prove that W/W_S is an upper bound on its speedup, no matter how many processing elements are used.*

Assignment 5. *(Prefix sums) Consider the problem of computing the prefix sums of n numbers on n processing elements. What is the parallel runtime, speedup, and efficiency of this algorithm? Assume that adding two numbers takes one unit of time and that communicating one number between two processing elements takes 10 units of time. Is the algorithm cost-optimal?*