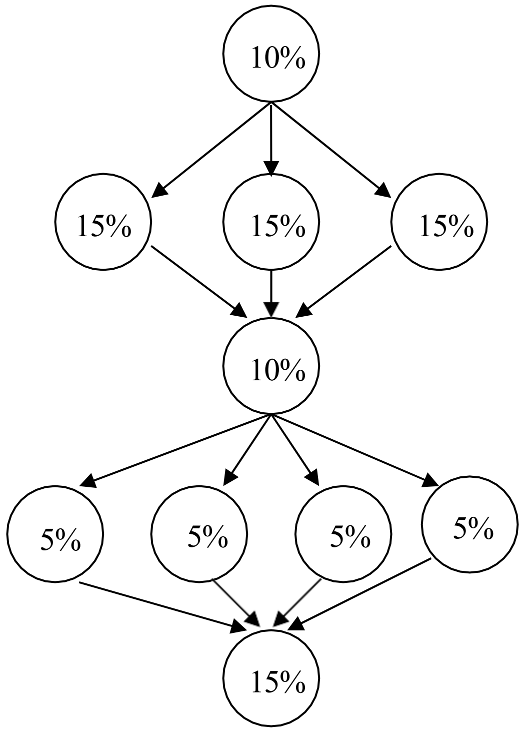
High Performance Computing
Parallel Computing

Problems Topic 1: Parallelism and Performance Evaluation

1.1.- The figure shows the network of dependencies between the tasks of an application. For each task the fraction of sequential execution time it takes is indicated. Assuming that the sequential execution time on a processor is 60 seconds and that the tasks cannot be divided into subtasks of lower granularity and there is no penalty for communications. We are asked to obtain the parallel execution time and the speedup obtained with:

- 4 processors.
- 2 processors.

Plot the parallelism profile of the program and the actual parallelism as a function of time achieved in each case.

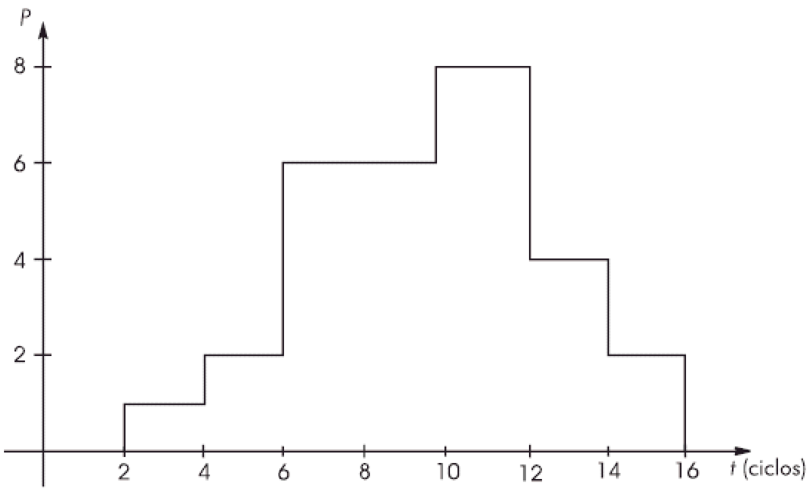


1.2.- 25% of a program cannot be parallelized, the rest can be distributed equally among any number of processors without considering overhead.

a) What is the maximum acceleration value that could be achieved by parallelizing it?

b) From what number of processors could accelerations greater than or equal to 2 be achieved?

1.3.- Let's suppose that a program has the parallelism profile shown in the figure:



a) Calculate the average parallelism of this program.

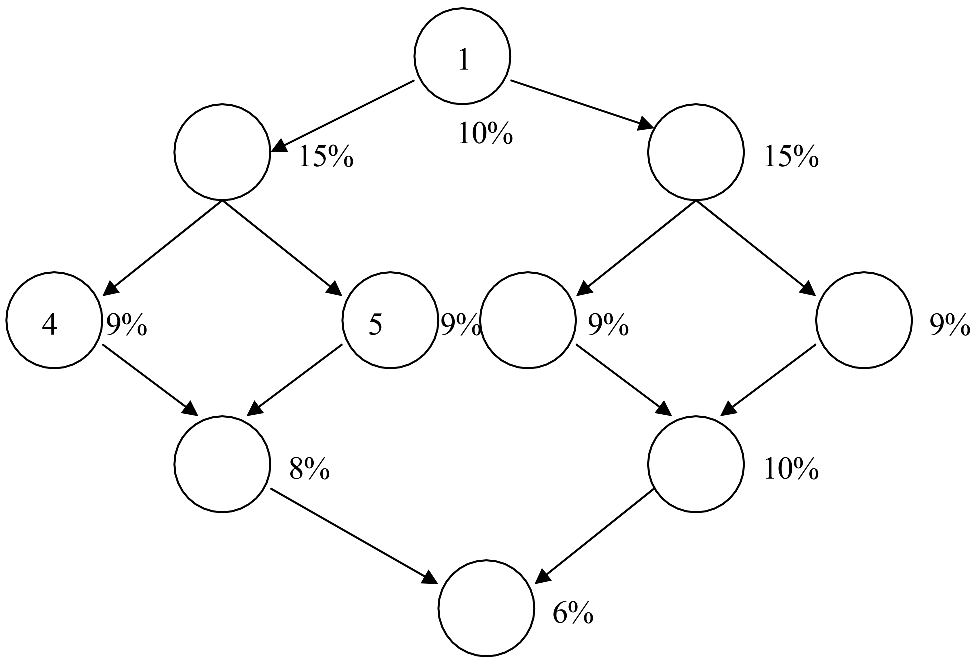
b) Calculate the workload of that process if the graph in the figure has been obtained with processors executing 2 instructions per cycle.

c) What is the maximum speed gain that can be achieved when running this program on a machine with 4 processors of the same speed?
d) Under the conditions of the previous sections: What would be the maximum speed gain that would be obtained, in that system, with a program having a constant parallelism degree of 8?

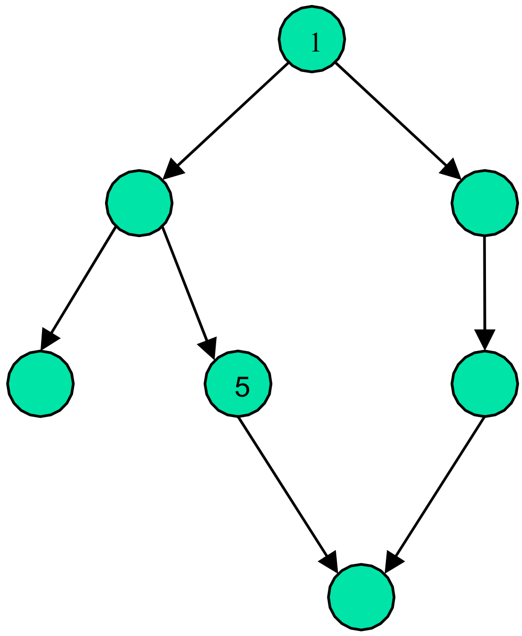
1.4.- A program, which took 2 seconds to execute in a processor, has been divided into 10 tasks. The order of precedence between the tasks is represented in the following figure, indicating also the percentage of time of each one. Considering that the communication time between tasks can be neglected.

Plot the parallelism profile of the program and calculate:

b) How long does it take to run the program in parallel?
?
b) What acceleration is obtained with respect to sequential execution?



1.5.- The execution of a program is distributed in seven tasks with identical computational cost and with the dependencies indicated in the following network.



In a system with 2 processors, the tasks are scheduled with the following distribution;

P0	1	
P1		5

Calculate

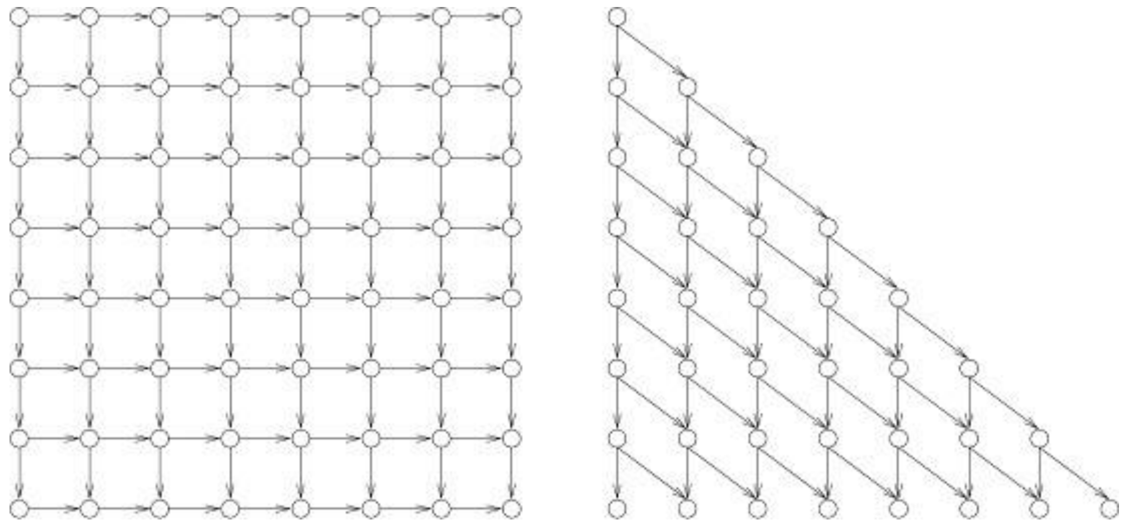
- Represent the program's parallelism profile
- Indicate the execution time on a single-processor system.
- The speedup achieved if you have two processors and use table scheduling.
- Is it possible to get more acceleration for two processors by varying the schedule? Which criterion improves planning?
- How many processors are needed to achieve the maximum possible acceleration.

1.6.- The efficiency of a system with n processors is 89%, solving a problem that initially is characterized by having a non-parallelizable 2%. Calculate the number of processors used in the system under the following assumptions.

a) The computational work is of fixed size and the n-processor system has allowed to reduce the execution time.

b) The computational work has been scaled with the number of processors to improve the accuracy of the solution.

1.7.- The figure represents the dependency network of an application to be executed in a multiprocessor system with p processors. If we generalize by calling N the number of nodes in a network and n is an integer, so that in the network there are N= n x n nodes in the figure on the left and N= n x (n+1)/2 nodes in the figure on the right.



It is assumed that there is no penalty for communications and that each node represents a task that takes one unit of time and cannot be divided into subtasks of lower granularity.

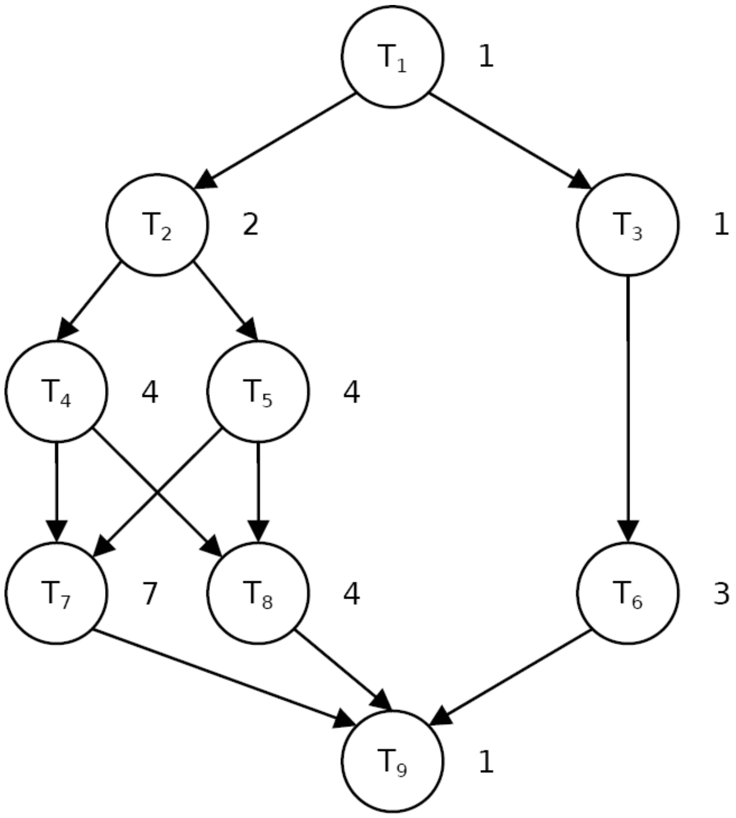
a) Plot the parallelism profile during the execution of the application for n=8

b) The maximum degree of parallelism and the work done with degree of parallelism three .

c) Calculate the speed-up (S) and efficiency (E) as a function of the number of processors S(p) and compare for this case the values S(), S(p=4) and S(p=8).

d) Express the maximum degree of parallelism and the maximum speed-up as a function of n.

1.8.- The figure represents the dependency graph of an application to be executed in a system with p processors. Each circle represents an independent task and the value to its right represents the number of cycles it takes (i.e.: T5 takes 4 cycles, T6 takes 3 cycles,...).



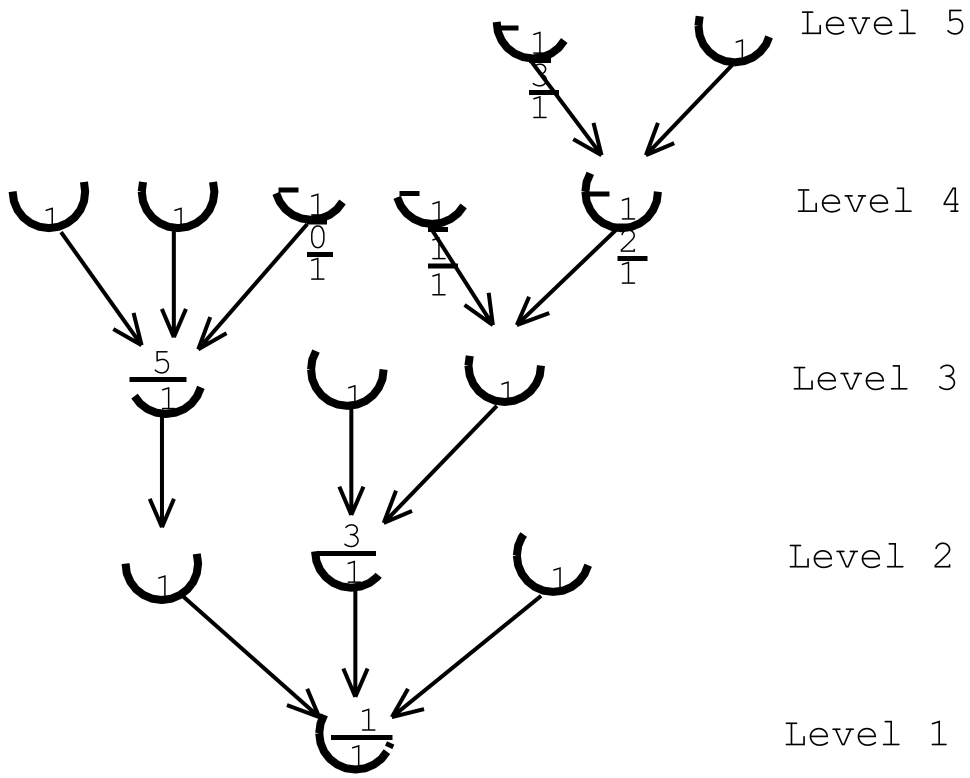
e) Plot the parallelism profile of this application.

f) Calculate the maximum degree of parallelism and the work done with degree of parallelism three.

g) Calculate the speed-up (S) as a function of the number of processors S(p) and compare for this case the values S(), S(p=2) and S(p=4).

h) Calculate the number of processors to maximize the efficiency of the system and then represent in a schedule the distribution of tasks as a function of time for each processor.

1.9.- The dependency graph of an algorithm is shown in the figure. Each task is executed in one cycle.



The scheduling scheme assigns each task a priority defined as the number of successors to each node and assigns each available processor the highest priority unexecuted ready task.

a) Indicate which critical path defines the longest planning length in time units.
b) Estimate the number of system processors that optimizes the execution of the algorithm.
c) Graph the planning result on a system with 3 processors, indicating the final execution time.
d) Repeat the previous section with 2 processors