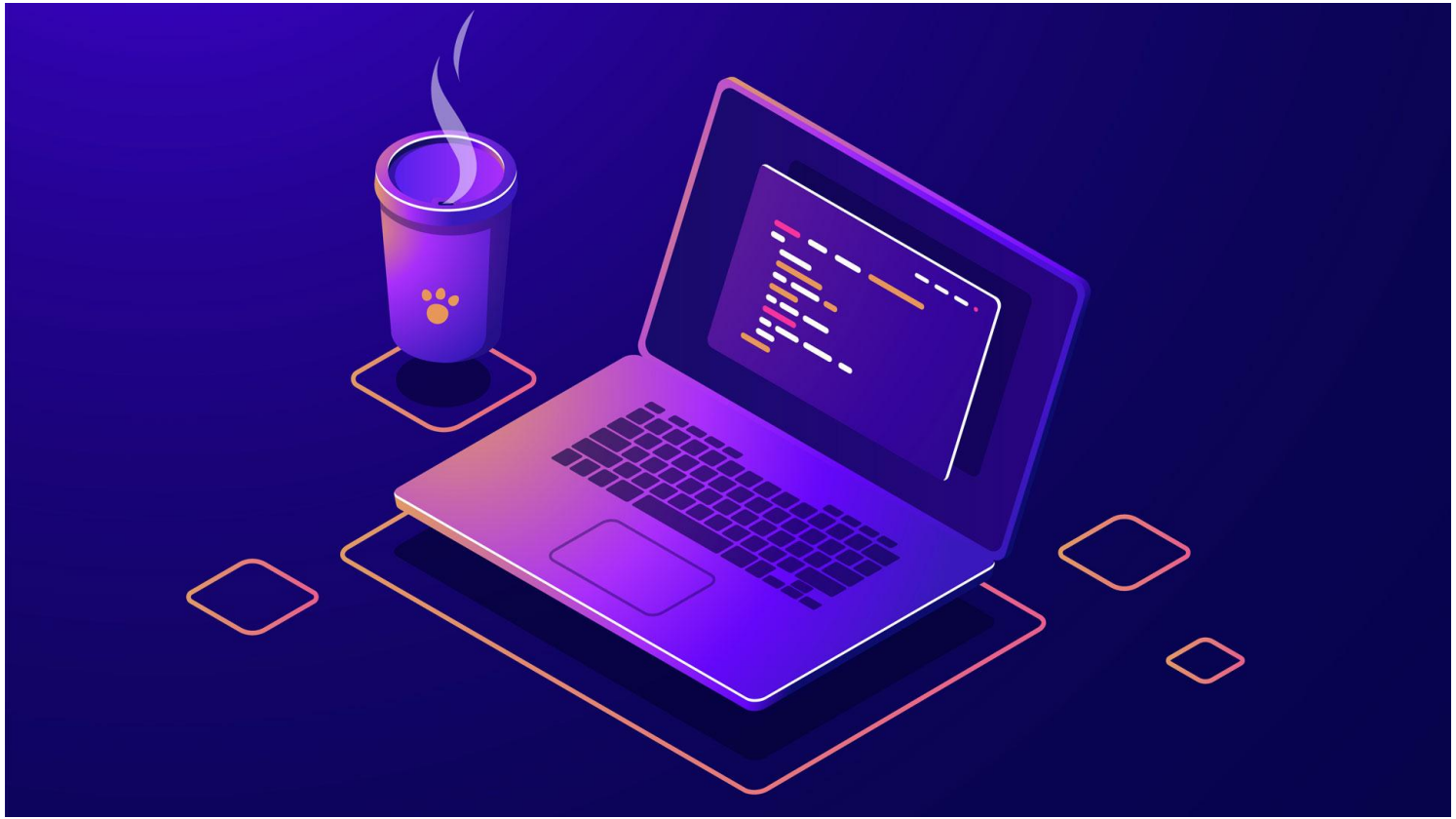
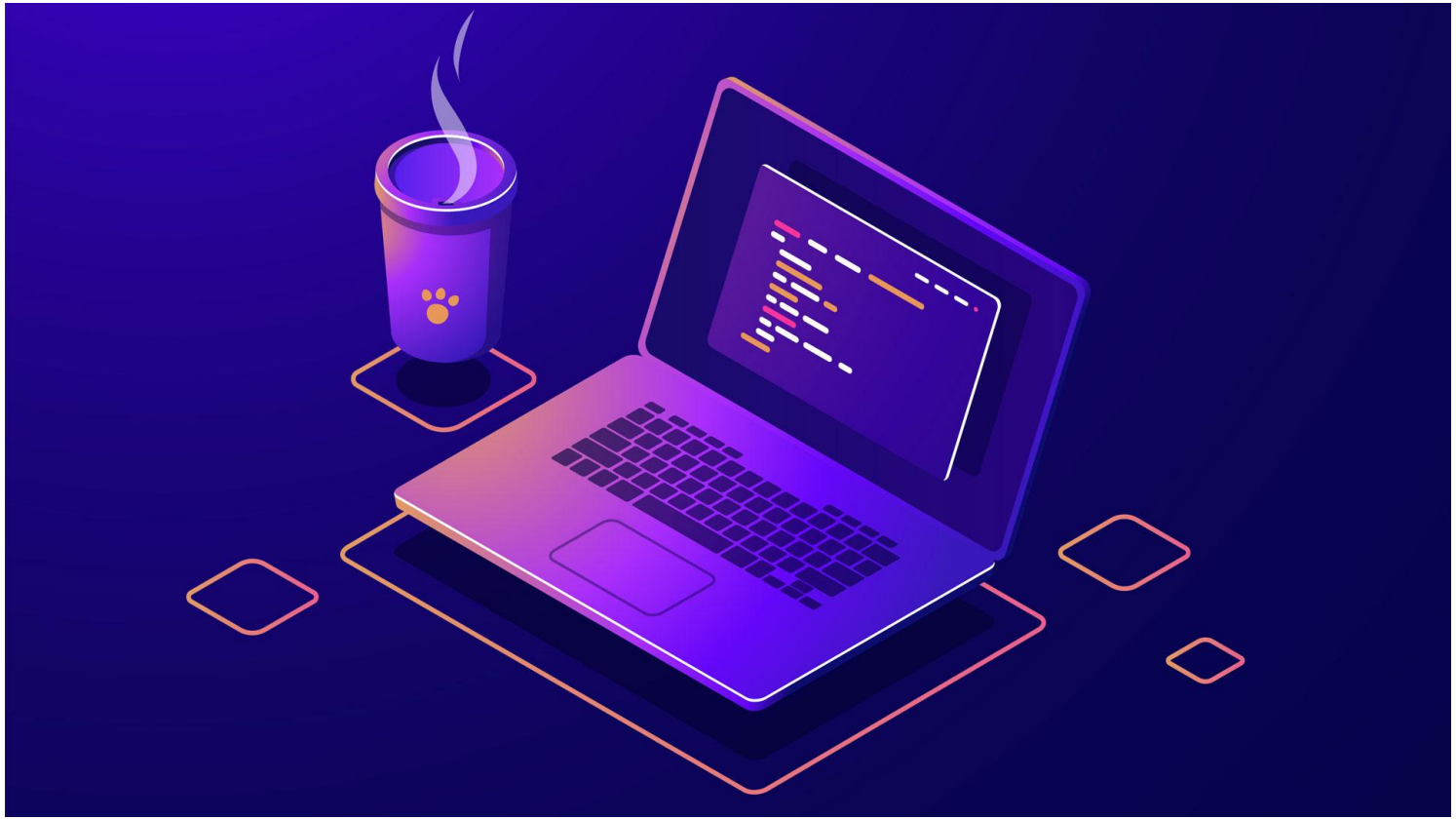
Parallelism Fundamentals Problems



Problem 3



```
# define MAX 8
// initialization
for (outer = 0; outer < MAX; outer++) {
     start_task(“for-initialize”);
     for (inner = 0; inner < MAX; inner++)
          matrix[outer][inner] = inner;
     end_task(“for-initialize”);
     }
// computation
for (outer = 0; outer < MAX; outer++) {
     start_task(“for-compute”);
     for (inner = 0; inner <= outer; inner++)
          matrix[outer][inner] = matrix[outer][inner] + foo(outer,inner);
     end_task(“for-compute”);
     }
```

Problem 3

Assuming that: 1) in the initialization loop the execution of each iteration of the internal loop lasts 10 cycles; 2) in the computation loop the execution of each iteration of the internal loop lasts 100 cycles; and 3) the execution of the **foo** function does not cause any kind of dependence. **We ask**:

a) Draw the Task Dependence Graph (TDG), indicating for each node its cost in terms of execution time.

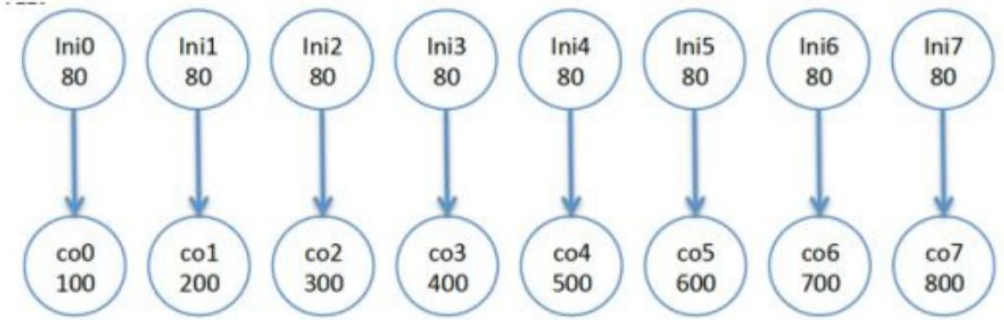
b) Calculate the values for T_1 and T_∞ as well as the potential *Parallelism*.

c) Calculate which is the best value for the ”speed-up” on 4 processors (S_4), indicating which would be the proper task mapping (assignment) to processors to achieve it.

Solutions

a)

Task Dependence Graph:



Solutions

b)

$T_1 = 4240$

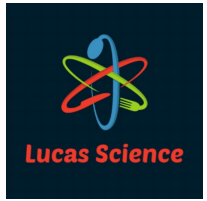
$T_\infty = 880$

Parallelism = $T_1 / T_\infty = 4.81$

So $P_{min} = 5$

Instructor Social Media

Youtube: Lucas Science



Instagram: lucaasbazilio

Twitter: lucasebazilio