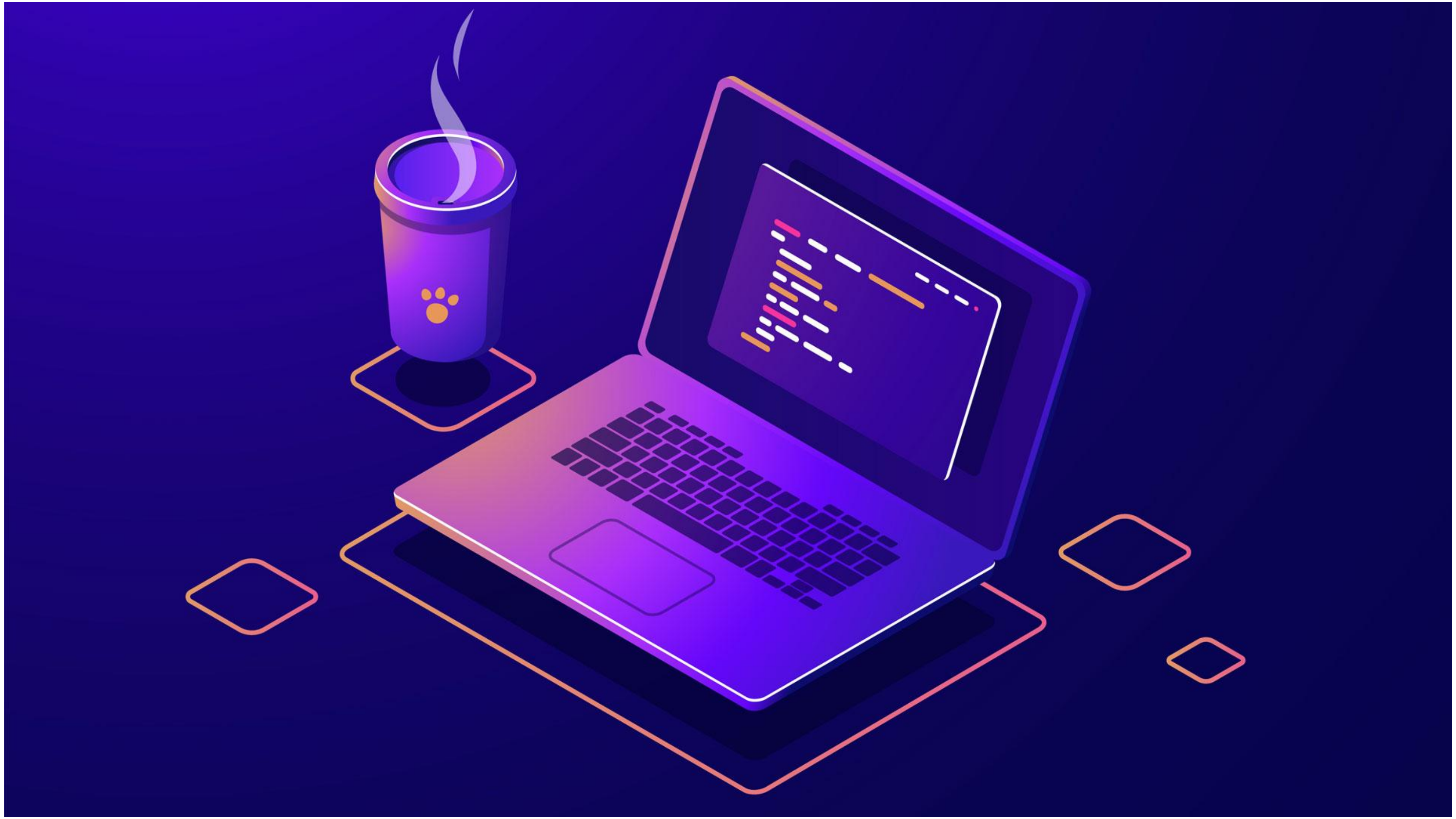
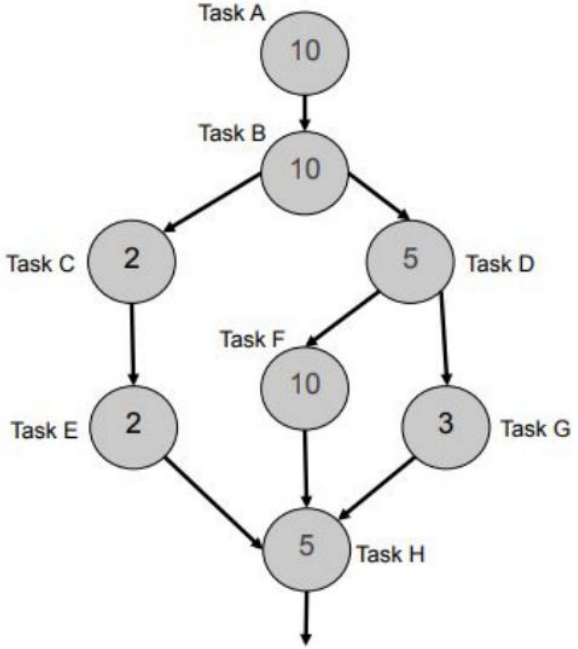
Task Dependency Graph

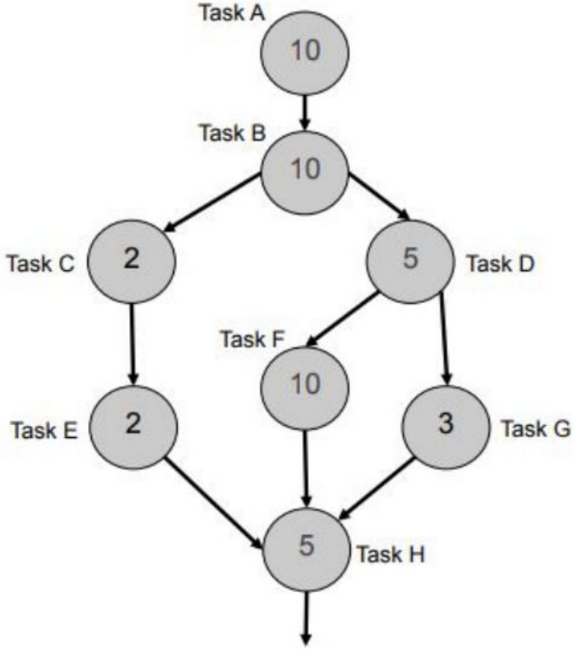


Task Dependency Graph



- **Task dependence graph** abstraction
  - Directed Acyclic Graph
  - Node = task, its weight represents the amount of work to be done
  - Edge = dependence, i.e. successor node can only execute after predecessor node has completed

Total Work T_1

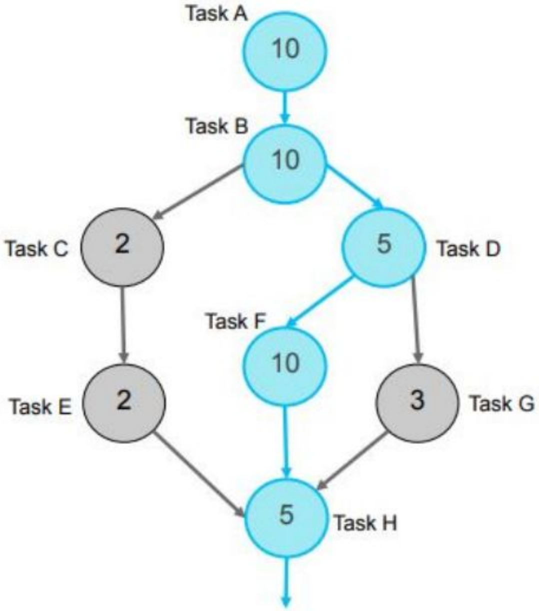


- Parallel machine abstraction
  - P identical processors
  - Each processor executes a node at a time

- $T_1 = \sum_{i=1}^{nodes}(work\_node_i)$

Lucas Bazilio - Udemy

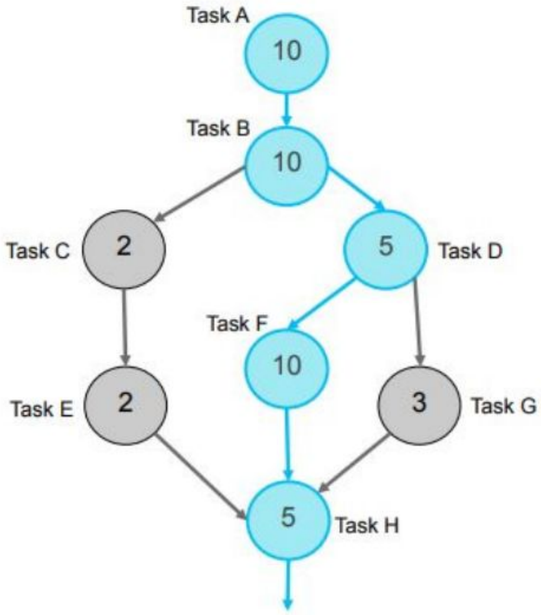
Critical Path Work T_∞



- Critical path: path in the task graph with the highest accumulated work
- $T_\infty = \sum_{i \in criticalpath}(work\_node_i)$, assuming sufficient processors

Lucas Bazilio - Udemy

Parallelism and P_{min}

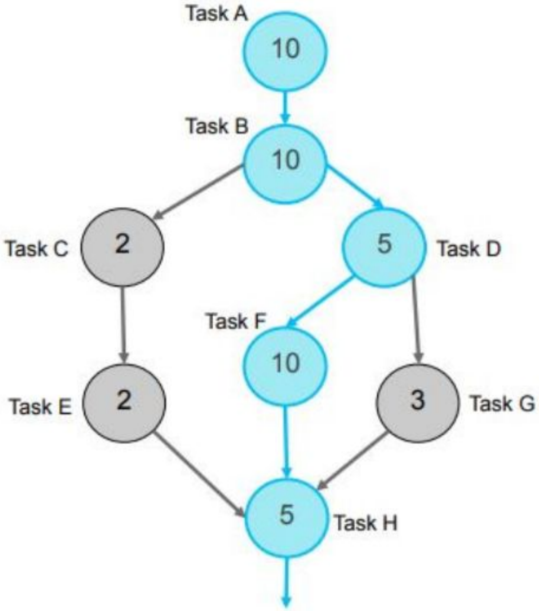


- $Parallelism = T_1/T_\infty$, if sufficient processors were available

- P_{min} is the minimum number of processors necessary to achieve $Parallelism$

Lucas Bazilio - Udemy

Wrap Up

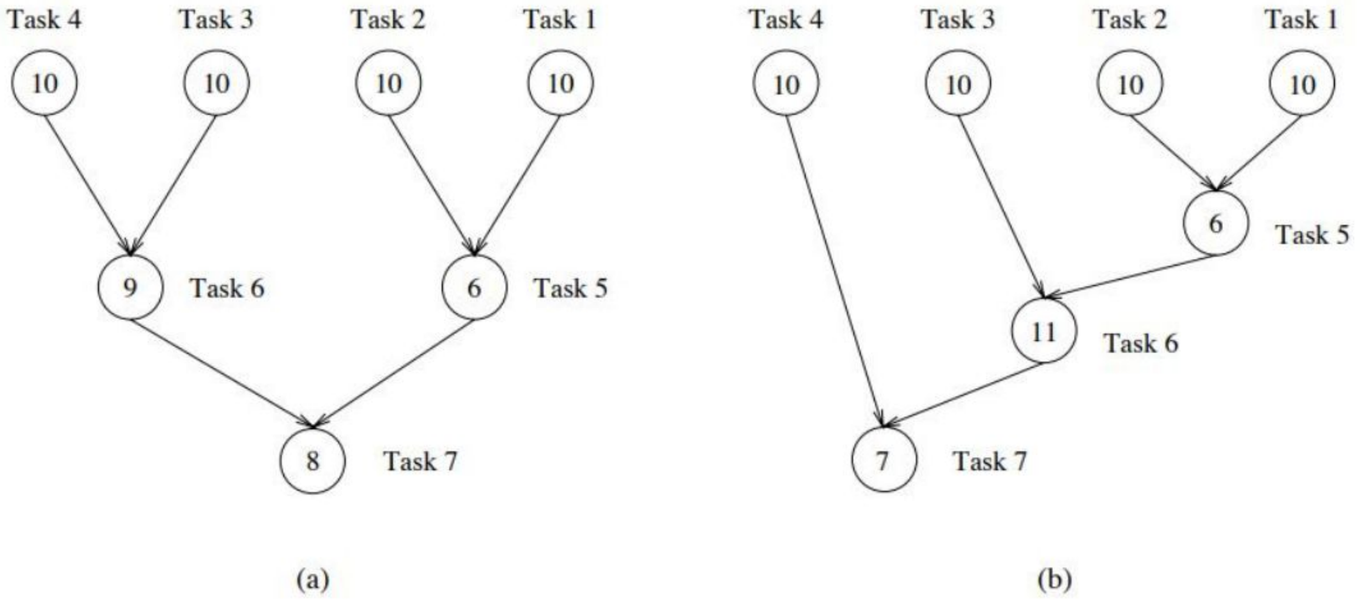


- $T_1 = 47$ including tasks {ABCDEFGH}
- Possible paths:

  {ABCEH} with total cost 29
  {ABDFH} with total cost 40
  {ABDGH} with total cost 33

- $T_\infty = 40$ for critical path {ABDFH}
- $Parallelism = 47/40 = 1.175$
- $P_{min} = 2$

Consider the task dependency graphs for the two database queries, assuming *work_node* is proportional to the number of inputs to be processed



Which are T_1, T_∞ and $Parallelism$ in each case?

Instructor Social Media

Youtube: Lucas Science

Instagram: lucaasbazilio

Twitter: lucasebazilio