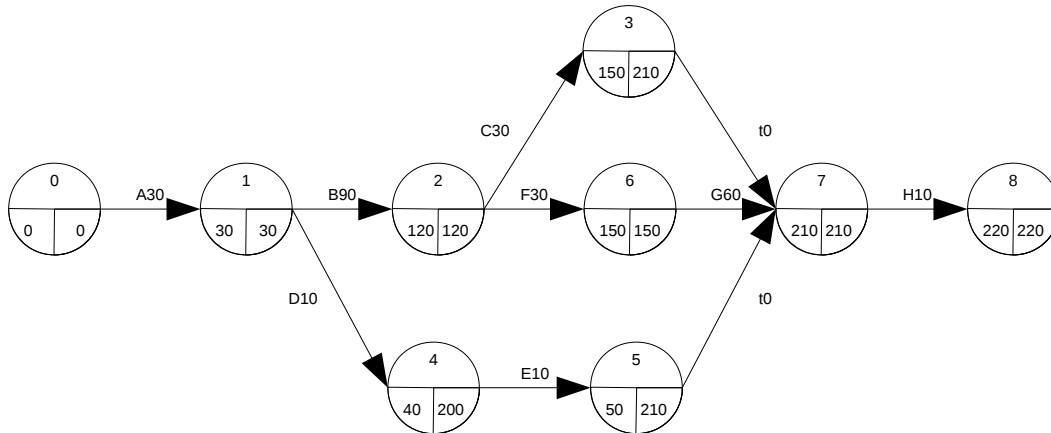


# TP DAG

- Proposez une structure de données complètes pour représenter un diagramme PERT. Les nœuds seront les étapes constituées d'un numéro, d'une date au plus tôt et d'une date au plus tard. Les arcs seront les tâches avec un nom et une durée, cf. exemple ci-après.



— La tâche « A » prend 30 unités de temps, la tâche « B » 90 unités de temps...

— L'étape 1 peut être passée au plus tôt après 30 unités de temps et doit être passée au plus tard après 30 unités de temps ; l'étape 4 peut être passée au plus tôt après 40 unités de temps et doit être passée au plus tard après 200 unités de temps...

- Écrivez les primitives suivantes pour manipuler votre diagramme PERT :
  - pour les étapes
    - get\_au\_plus\_tot
    - get\_au\_plus\_tard
    - get\_number
    - get\_next\_steps
    - get\_previous\_steps
  - pour les tâches
    - get\_name
    - get\_duration
    - get\_begin\_step
    - get\_end\_step
- Réalisez une fonction « critique » qui prend en paramètre un diagramme PERT et qui retourne le chemin critique.
- Étant donné un diagramme PERT pour lequel les dates au plus tôt et au plus tard n'ont pas été renseignées :
  - implémentez la méthode « compute\_au\_plus\_tot » qui calcule, pour chaque étape du PERT, les date au plus tôt.

2. implémentez la méthode « `compute_au_plus_tard` » qui calcule, pour chaque étape du PERT, les date au plus tard.