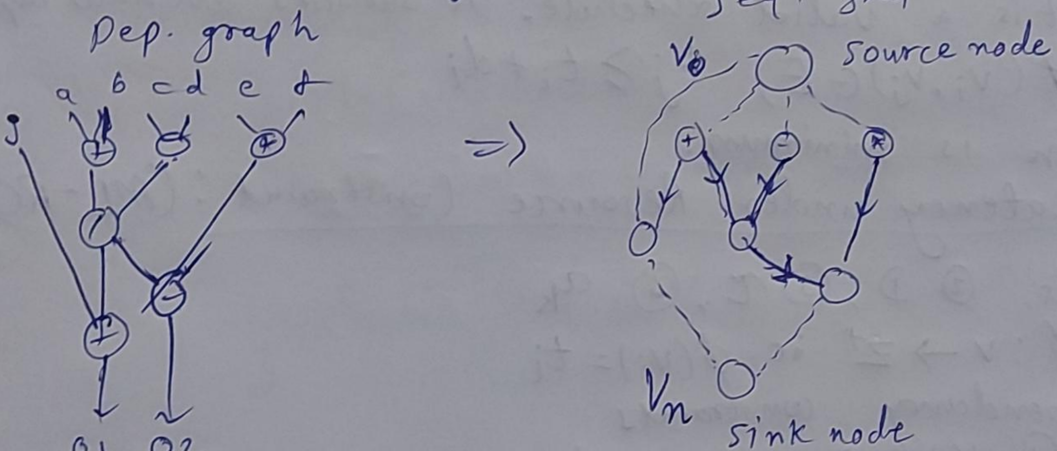
Scheduling:

- Assign timestamp to each operation.
- Determine ~~redundancy~~ the latency of your design.

Inputs: ① Dependency graph $G = (V, E)$, V = operations, E = dependency (v_i, v_j)



② $D = \{d_1, d_2, \ldots d_n\}$, d_i is the delay of of v_i.

③ $T : V \rightarrow \{1, \ldots n_{res}\}$
n_{res} is types of operation.

④ Resource constraints: $a_k : k = 1, 2 \ldots n_{res}$

⑤ Latency constants : λ

Outputs: ① $T : \{t_i \mid i = 0, \ldots n\} \Rightarrow \phi : V \rightarrow \mathbb{Z}^+ \; \phi(v_i) = t_i$

t_i represents the schedule (start time) of node v_i.

② $(t_n - 1)$ is the latency. $t_0 = 0$.

Unconstraint scheduling

Given: ① G ② D ③ τ

find: $\phi : V \to Z^+$

~~s.t.~~ ② $\phi(v_i) = t_i$

s.t. ① It is a valid schedule, it satisfies all data dep.

$\forall (v_i, v_j) \in E, \; t_j \geq t_i + d_i$

② t_n is minimum.

Minimize Latency under Resource Constraint: (ML-RC)

Given: ① G, ② D, ③ τ, ④ a_k

Output: $\phi : V \to Z^+$ i.e. $\phi(V_i) = t_i$

s.t. ① Dependency constraints

$\forall (V_i, V_j) \in E, \; t_j \geq t_i + d_i$

② $\{V_i : \tau(V_i) = k$ and $t_i \leq l \leq t_i + d - 1\} \leq a_k$

for each $k = \{1, 2, \ldots n_{res}\}$ and

for each time step $l = 1, 2, \ldots t_n$

$\Rightarrow$ Resource Constraints must be satisfied.

③ t_n is minimum.

Minimize Resource under Latency Constraint: (MR-LC)

Given: ① G, ① D ③ τ ④ λ (latency limit).

output: $\phi : V \to Z^+$ ie $\phi(v_i) = t_i$

s.t. ① Dep. constraint

$\forall (v_i, v_j) \in E, \; t_j \geq t_i + d_i$

② Latency constraint

$t_n \leq \lambda + 1$

③ $a_1, a_2, \ldots a_{res}$ is minimum.

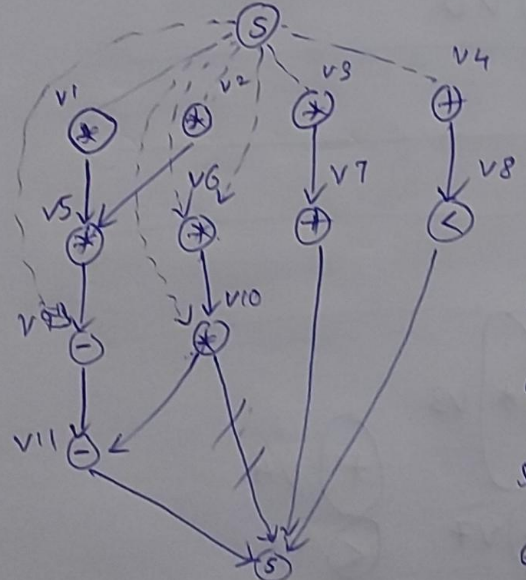
$\boxed{\left.\begin{array}{l} \text{ML-RC} \\ \& \\ \text{MR-LC} \end{array}\right\} \text{NP-complete} \quad \text{but US not NPC.}}$

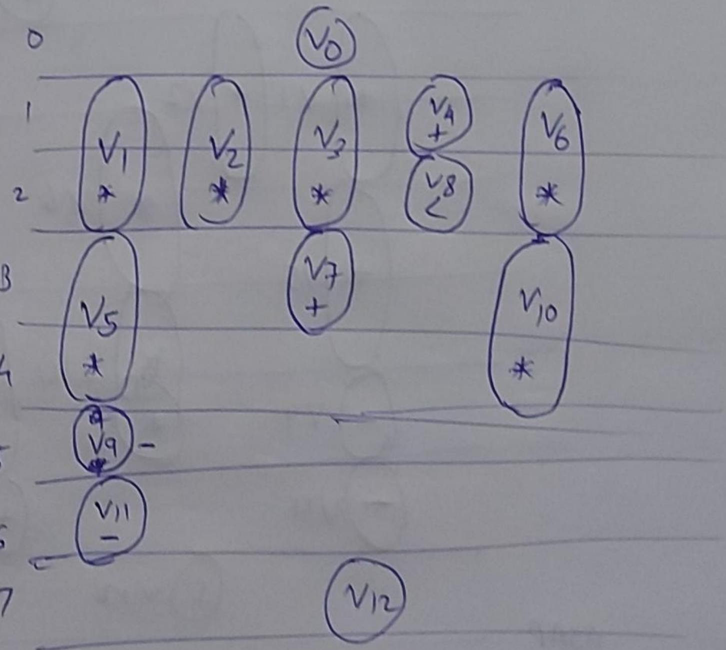
Unconstraint Scheduling:

① ASAP (As Soon As possible)

→ schedule each node ~~ASAP~~ as soon as all predecessors nodes completes the execution.

ASAP (G, D, τ)
{
    ~~$Ts_0 = V$~~, $t_0 = 0$,
    while (unscheduled node exists)
    {
        1. select a node v_j where predecessor have already been scheduled.
        2. schedule node v_j to t_j where
           $t_j = \max(T_i + d_i) \ \forall (v_i, v_j) \in E$.
    }
}





$d(*) = 2$

$d(+, -, <) = 1$

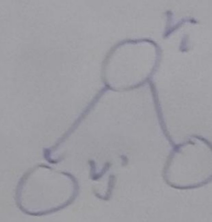
$\lambda = 6$

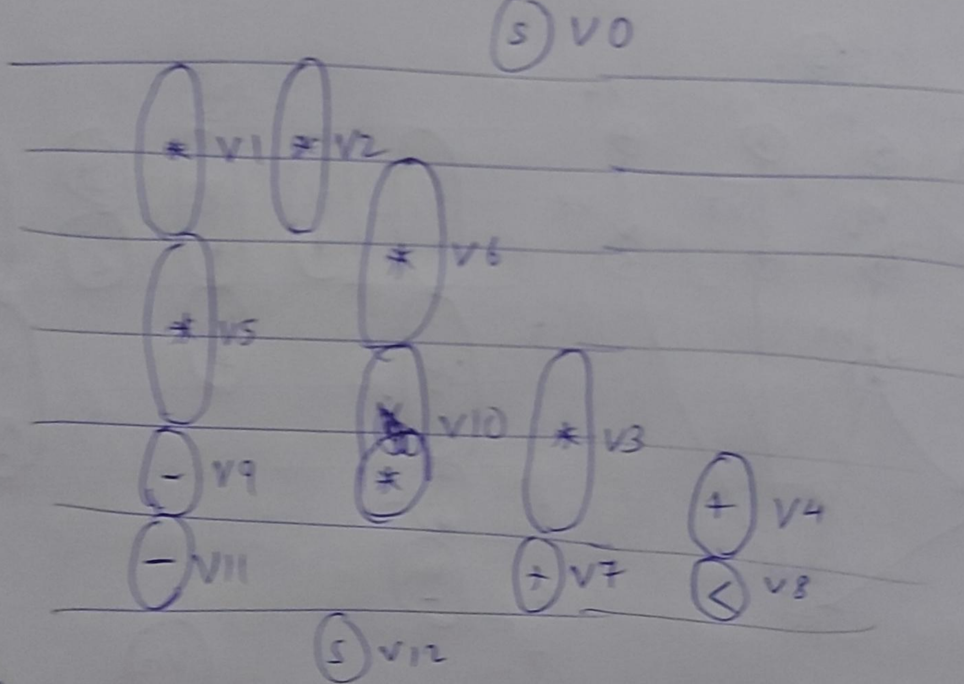
Sequence Graph of Diffeq.

② ALAP (As Late As Possible)

ALAP(G, D, τ, λ) ↳ latency bound
{
$t_n = \lambda + 1$
while (any node available)
{
1. Select a node V_i whose successors are already scheduled.
2. Schedule V_i to $t_i = \mathrm{Min}(t_j - d_i)$ $\forall (v_i, v_j) \in E$

}
if ($t_0 < 0$) "λ is not sufficient" Error;
}





t_i^{ASAP} : Lower bound of schedule of node V_i

t_i^{ALAP} : Upper " " " " "

Mobility of node V_i = $t_i^{ALAP} - t_i^{ASAP}$

ILP Formulation of Scheduling
(Integer linear programming)

$x_1, x_2, x_3, \ldots x_n$: unknown Int var

Constraint

$e x_1 + x_2 + 5x_3 \leq 10$

obj. fun: minimize $x_1 + x_2 + x_3 \ldots$

obj. $\min c^T x$

s.t. $Ax \leq b$

$\Rightarrow x_{il} = \begin{cases} 1, & \text{if } v_i \text{ scheduled in time step } l \text{ (start time)} \\ 0, & \text{otherwise} \end{cases}$

- For diffeq. $|V| = 11$: 66 variables
  $\lambda = 6$

$\rightarrow x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} = 1$ (unique start time)

$\sum_{l=1}^{\lambda} x_{il} = 1, \forall i = 1, \ldots n$

$\rightarrow t_i = \sum_l l \cdot x_{il}$