- Exam 2 : Friday, March 24.
  - Same policies as Exam 1.

- memorizing will not help much.

- recreate the material - lectures, homeworks, rec.
  - ↳ write down the details as if explaining the concept/problem to someone else.

- Topics
  - Asymptotic notation
  - Stable matching
  - Divide & Conquer

- Things to keep in mind.
  - when we say "No partial credit will be given for incorrect answers", do not waste your time giving justification.

- You do not need to justify anything that is already covered in lecture, homework, recitation.

  Ex: The solution to the recurrence

  $$T(n) = T(n/2) + c \quad \text{and } T(1) = 1$$

  is $T(n) = \Theta(\lg n)$

  This was done in class, so you can use it unless we explicitly ask you to justify.

- When asked for a runtime recurrence, you must always give the base case, otherwise you will lose points.

- Use Simplified Master Theorem whenever you can.

- Cheat Sheet will be provided with some formulae & the Simplified Master Theorem.

- Next meeting : Mon, March 27.

Asymptotic Notation.

O , Ω , Θ , o , ω

O → asymptotic upper bound

Ω → asymptotic lower bound

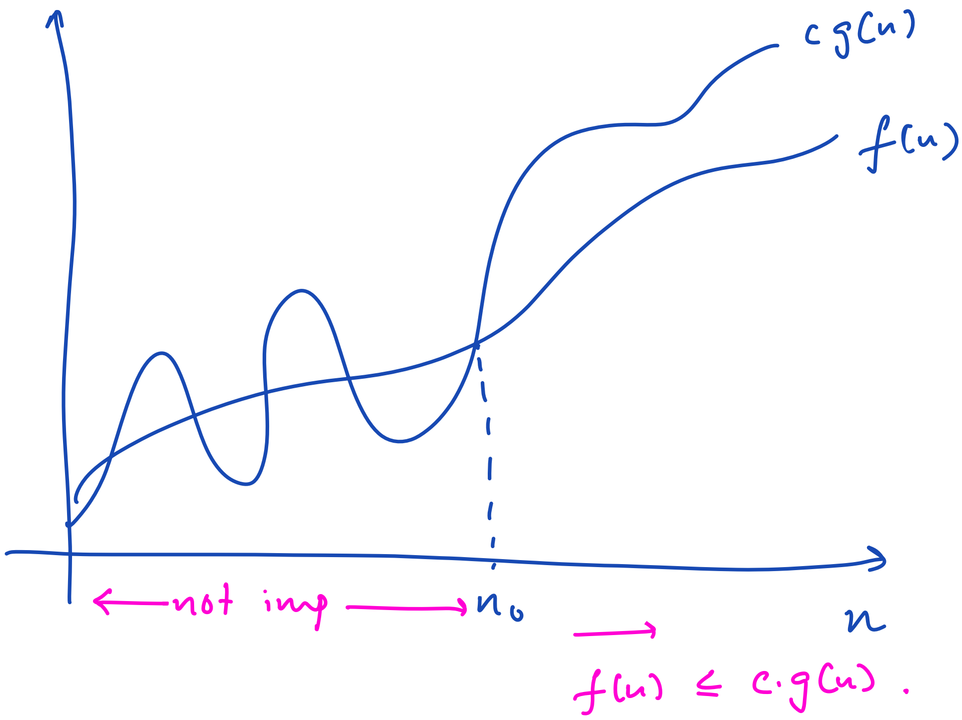
Θ → tight bound.

o, ω — will not test

"Big-O" O.

$O(g(n)) = \{ f(n) \mid \exists$ positive constant c & n_0

s.t. $\forall n \geq n_0$, $\boxed{0 \leq f(n)} \leq c g(n) \}$



$f(n) \in O(g(n))$

$f(n) = O(g(n))$.

Ex : $\boxed{7n - 10} = O(n)$

$\underline{7n - 10} \le 7n \quad , \quad \forall \; \underline{n} \ge \underline{2}$

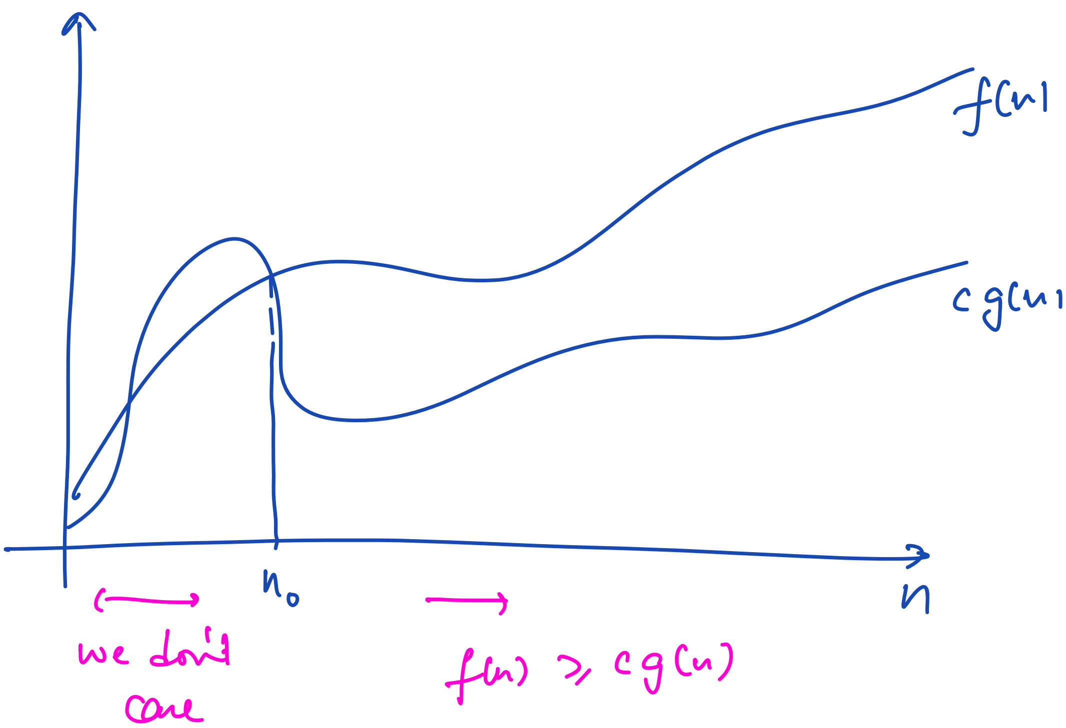
1

$c = 7$, $n_0 = 2$.

Big-Omega

$\Omega(g(n)) = \{ f(n) \mid \exists$ positive consts c & n_0,

s.t. $f(n) \geq c\,g(n) \geq 0$, $\forall n \geq n_0 \}$



Ex: $\underbrace{10n^3 + 30n^2 + n}_{f(n)} = \Omega(n^3)$

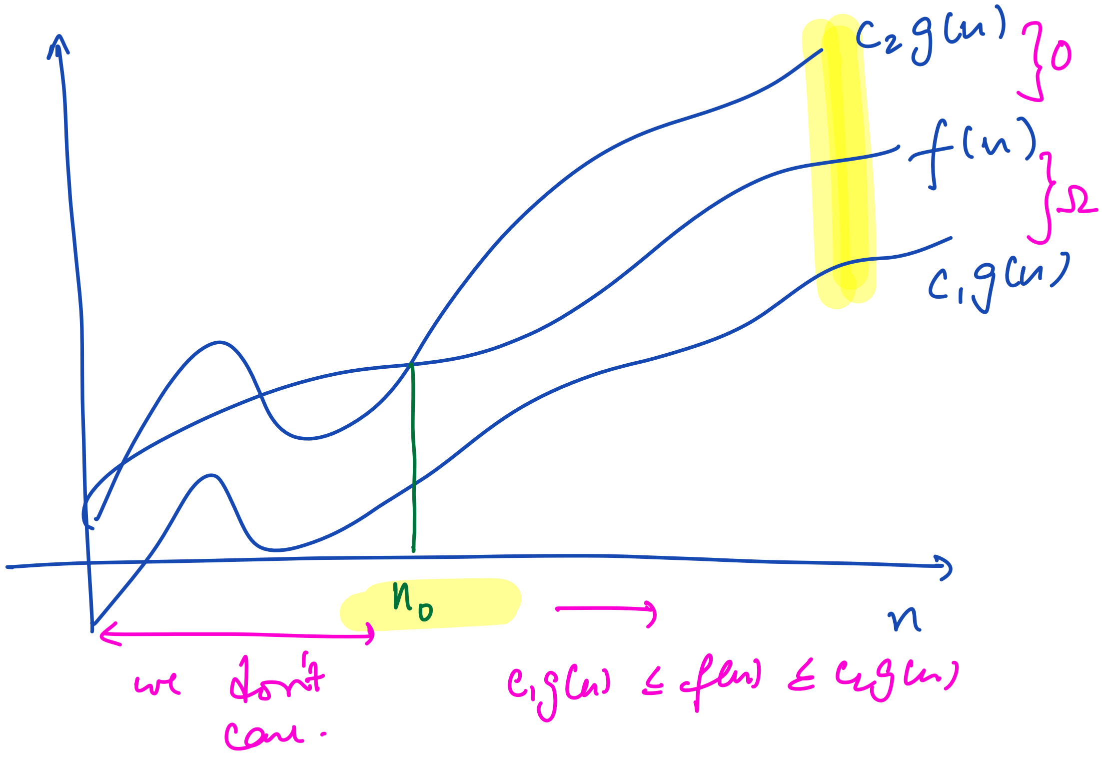
$10n^3 + 30n^2 + n \geq 1 \cdot n^3$, $\forall n \geq 1$

$c = 1$, $n_0 = 1$

$c = 10$, $n_0 = 1$

$\Theta(g(n)) : \{ f(n) \mid \exists$ positive consts c_1, c_2, n_0

s.t. $\forall n \geq n_0$, $c_1 g(n) \leq f(n) \leq c_2 g(n) \}$



Ex : $10n^2 + 33n = O(n^2)$ X false

false

True $\leftarrow$ $10n^2 + 33n = \Omega(n^2)$ ✗ false

false $\leftarrow$ $10n^2 + 33n = \Theta(n^2)$ ✓ True

$10n^2 + 33n \leq 1000 \cdot n^2, \quad \forall n \geq 1.$

$c = 1000, \quad n_0 = 1$

$o \begin{cases} O \\ \neq \Theta \end{cases} \qquad \omega \begin{cases} \Omega \\ \neq \Theta \end{cases}$

$n = o(n^2)$

$n \leq n^2, \quad \forall n \geq 1.$

$c = 1, \quad n_0 = 1$

$\Rightarrow \quad n = O(n^2)$

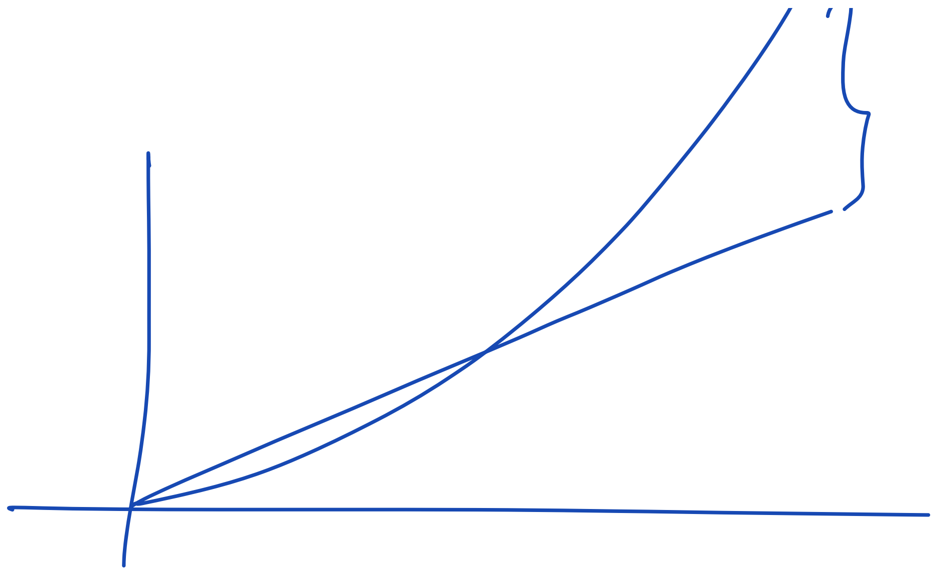
Clearly, $n \neq \Omega(n^2)$.

$\therefore \quad n \neq \Theta(n^2)$

$\therefore \quad n = o(n^2)$.

$$\left.\begin{array}{l} n^2 = \omega(n) \\ n^2 = \Omega(n) \\ n^2 \neq \Theta(n) \end{array}\right\} \Rightarrow n^2 = \omega(n).$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \begin{cases} 0, & f(n) = o(g(n)) \\ \infty, & f(n) = \omega(g(n)) \\ c, & f(n) = \Theta(g(n)) \end{cases}$$



$f(n)$

$10n^2 + 33n + 500$

$g(n) = n^2$

$f(n) = O(g(n))$, $f(n) = \Omega(g(n))$,

$f(n) = \Theta(g(n))$,

$f(n) = n^2$

$g(n) = \lg n$

$f = O(g)$ $\quad f = \Omega(g),\ f = \Theta(g)$

Ex: $\frac{n^2}{8} - 50n = \Theta(\underline{\ n^2\ })$

$\frac{n^2}{8} - 50n \leq c_2 \cdot n^2$

$c_2 = \underline{100}, \quad n_0 = 400$

$\frac{16 \cdot 10^4}{8} - 20000 > 0$

$$\frac{n^2}{8} - 50n \le 100n^2, \quad \forall n \geq 400.$$

$$\therefore \frac{n^2}{8} - 50n = O(n^2).$$

$$\frac{n^2}{8} - 50n \geq c_1 \cdot n^2, \quad \forall n \geq n_0.$$

$$\frac{n}{8} - 50 \geq c_1 n$$

$$\frac{n}{8} - c_1 n \geq 50$$

$$n\left(\frac{1}{8} - c_1\right) \geq 50$$

let $c_1 = \frac{1}{16}$

$$\therefore n\left(\frac{1}{16}\right) \geq 50$$

$$n \geq 800.$$

$c_1 = \frac{1}{16}$, $n_0 = 800$.

$$\frac{n^2}{8} - 50 = \Omega(n^2)$$

$$\therefore \frac{n^2}{8} - 50 = \Theta(n^2).$$

$$\lim_{n \to \infty} \frac{\frac{n^2}{8} - 50n}{n^2} = c.$$

$f(n) = (\underline{\lg n})^{\underline{100}}$

$g(n) = \underline{n}^{0.1}$.

$f(n) = O(g(n))$. ✓

DFS :

DFS (G)

$O(n)\left\{\begin{array}{l}\text{for each } u \in V \text{ do} \\ \quad \text{color}[u] \leftarrow \text{white} \\ \quad \pi[u] \leftarrow \text{NIL}\end{array}\right.$ // π : parent printer

time ← 0

for each $u \in V$ do

if color[u] is white then

DFS_VISIT (u)

DFS_VISIT (u)

color[u] ← Gray

time ← time + 1

d[u] ← time

$\sum_u dg(u) = O(m)$

for each $v \in N(u)$ do

if color(v) is white then

"hi"

$\pi[v] \leftarrow u$

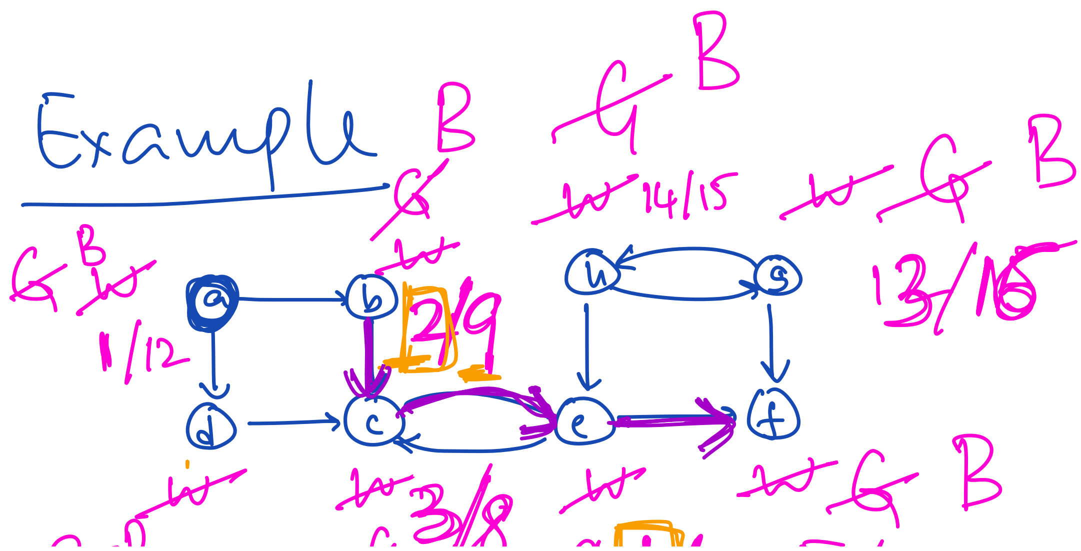
DFS_VISIT (v)

color[u] ← Black

time ← time + 1

f[u] ← time.

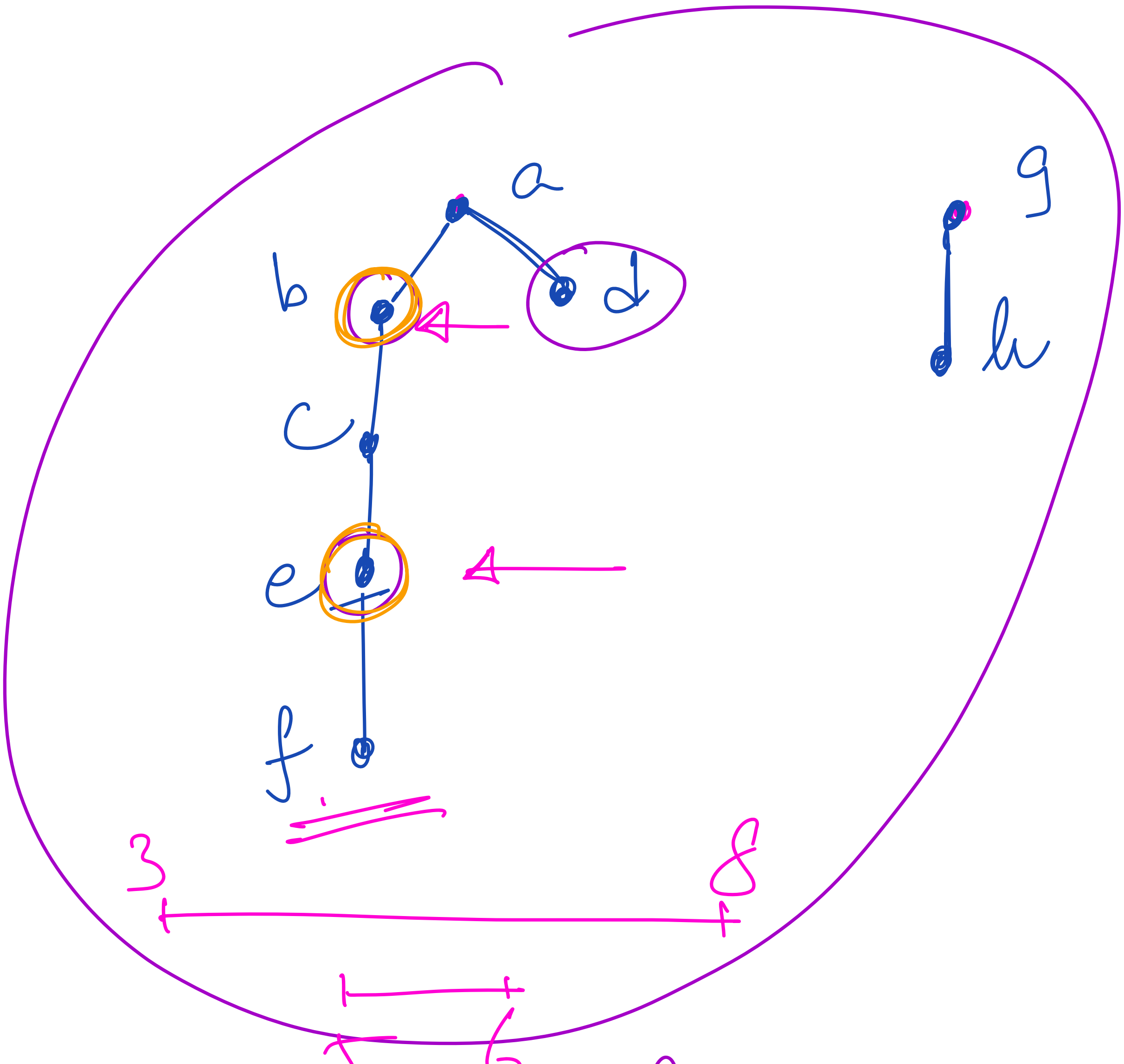
Example



~~6~~ B B ~~8~~ 10 B 4/7 5/6

10/11

time ← ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~ ~~11~~ ~~12~~ ~~13~~ ~~14~~ 15



DFS forest

Running time : $O(n+m)$.

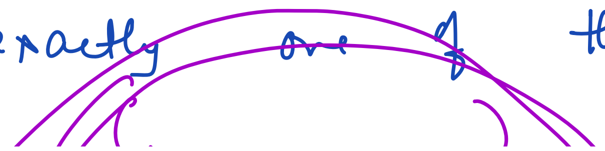
Properties of DFS

When is a vertex v a descendant of vertex u in the DFS forest?

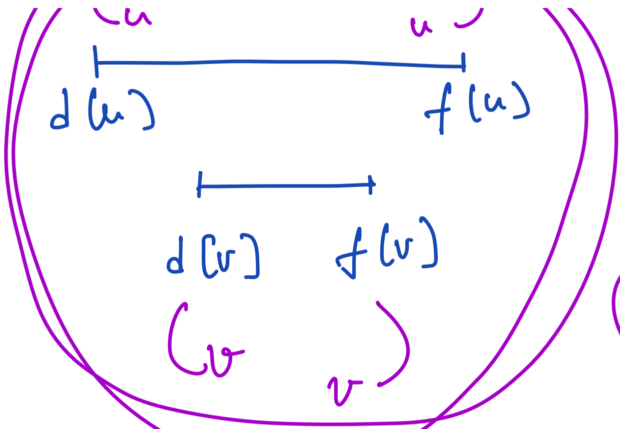
1. v is a descendant of u iff v is discovered when color[u] is gray.

2. **Parenthesis theorem.**

For any two vertices u & v in G, exactly one of the following happens.



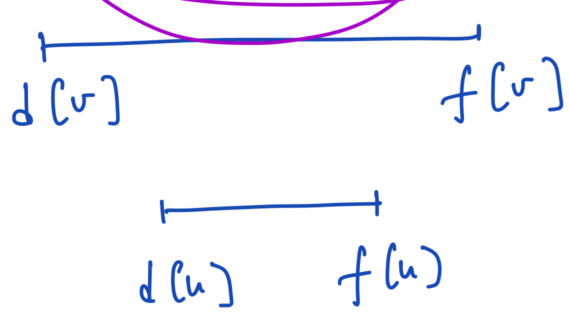
(a)



& v is a descendant of u in the DFS forest

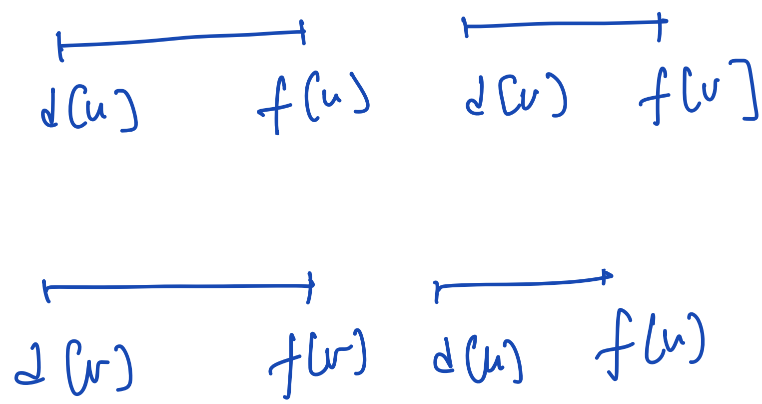
$(_u \; (_v \; {}_v) \; {}_u)$

(b)

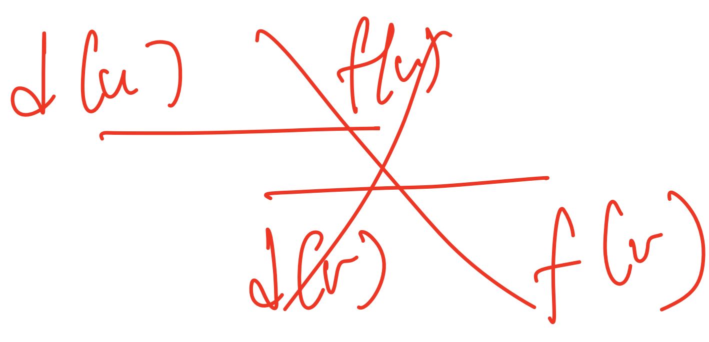


& u is a descendant of v in the DFS forest.

(c)



& neither u nor v are descendants of the other.



Corollary: v is a descendant of vertex u iff

$$d_u < d_v < f_v < f_u.$$

3. White Path Theorem

Vertex v is a descendant of vertex u iff at time $d(u)$ there is a path consisting only of white vertices from u to v in G.