Integer KP vs Fractional KP.

- Perbedaan :

$\{x_1, x_2, \ldots, x_n\}$; n : #item

↓ ↓ ↓

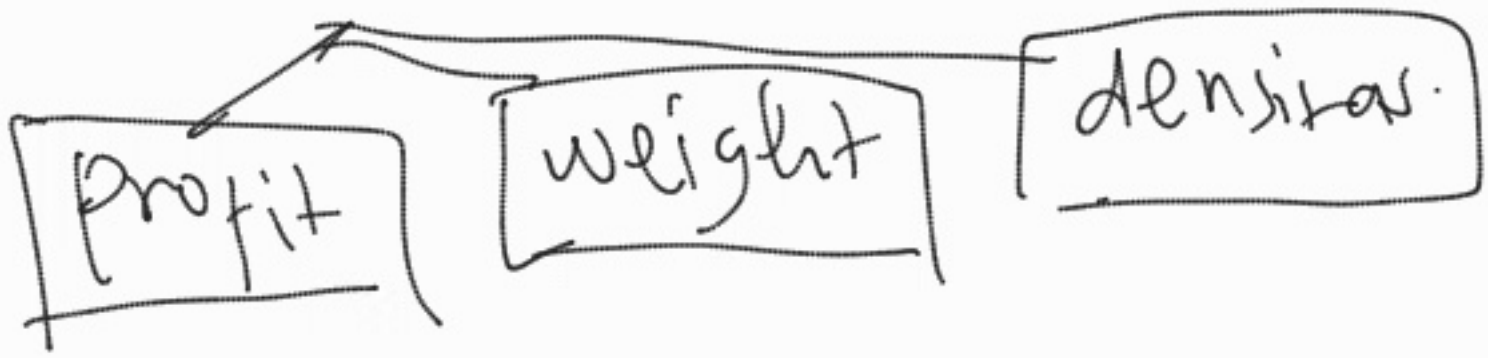
Int. $\in \{0, 1\}$

$x_i = 0$ atau $x_i = 1$

↓ ↓

objeknya tdk diambil diambil.

- Greedy Int :



	Pi	Wi	Pi/Wi
1	12	4	2 ✓
2	15	5	3 ✓
3	50	10	5 ✓
4	10	5	2

K = 16

→ dem 1
totwt = 15

Wt = 15

Wt = 16

{3, 2}

{2, 4, 1}

- Profit : 3, 2, 1, 4 ↘
- weight : 2, 4, 1, 8 ↗
- density : 3, 2, 1, 4 ↘

- Mengurutkan / presorting objek by (pr, wg, dens.)
- for i = 1 to n. (obj. yg diinput).
  - if sum + wi < K
    - indxde objek i

Masalah optimisasi

↳ Integer solution

↓ Relaksasi kondisi

$$0 \leq x_i \leq 1.$$

Masalah opt → lebih mudah.

$$\begin{cases} \max \sum_{i=1}^{n} p_i x_i \\ \text{- st. } \sum_{i=1}^{n} w_i x_i \leq K. \\ 0 \leq x_i \leq 1 \quad \forall i \end{cases}$$

↳ Pemrograman linear/linier.

Fractional

total wt = k

kita boleh mengambil sebagian

Fract. KP by density

memberi solusi opt.

Pseudocode Fractional KP.

- Presorting (mengurutkan objek). → $\Theta(n \log n)$

```
FractKP ( objek yg terurut : C )
                sesuai density.

for i = 1 to n.
    └ xi = 0.                    ∈ semua belum diambil.
                , profit ← 0.
i ← 1        totwt ← 0        intFrac ← T
while i ≤ n. & intFrac = T → O(n).
    if totwt + wi ≤ K.                    } O(1)
        └ xi ← 1                          profit ← profit + pi
    else ( totwt + wi > K.
        └ intFrac ← F
            xi ← (K − totwt) / wi
                                          profit ← profit + xi pi
return x, profit.
```

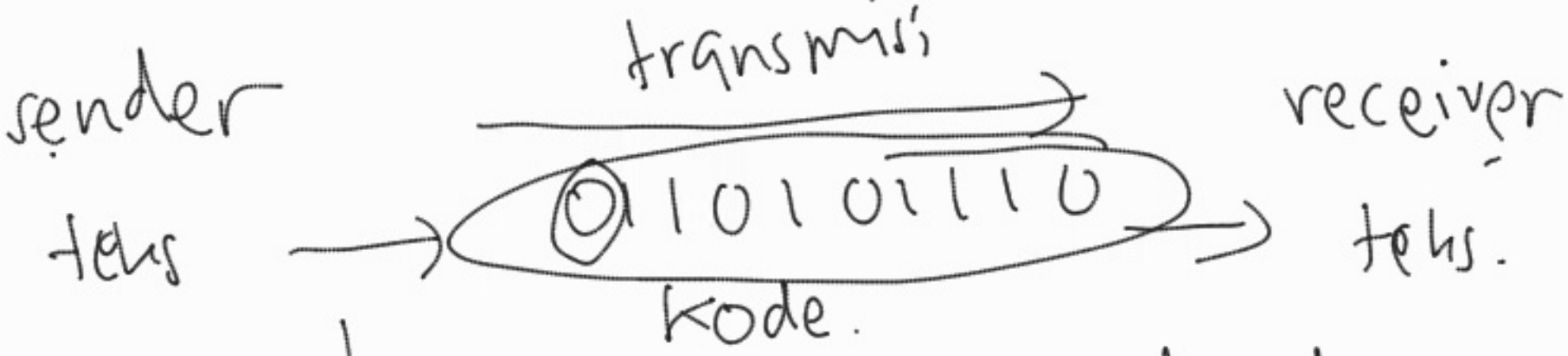
O(n)

Algoritma Huffman

Coding / Pengkodean ← Coding theory

Binary Code ← bit {0, 1}

0 1 1 0 1 0 1 . . . . .



- pesan tdk berubah / minimumkan error
- pesan sampainya cepat

Encoding

- Fixed-length
- Variable-length

a, b, c, d, e, f, ~~g~~ h i j ~~z~~ ...

a 000
b 001
c 01
d 100
e 101
f 111

Variable-length → Kode Huffman

frekuensinya besar

root

a : 45

c : 12

b : 13

f : 5

e : 9

d : 16

c : 2

d : 3

frekuensinya kecil.

Pengantar Graf

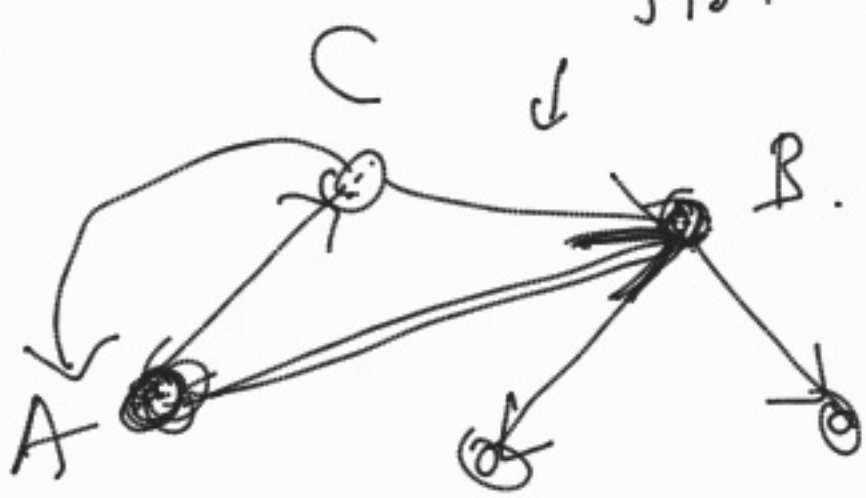
1) Minimum Spanning Tree.
2) Shortest Path.

Graf : titik + sisi
(vertex/node) (edge).

tdk berarah (undirected)
relasi ↕ 2 arah

berarah (directed).
relasi ↓ 1 arah

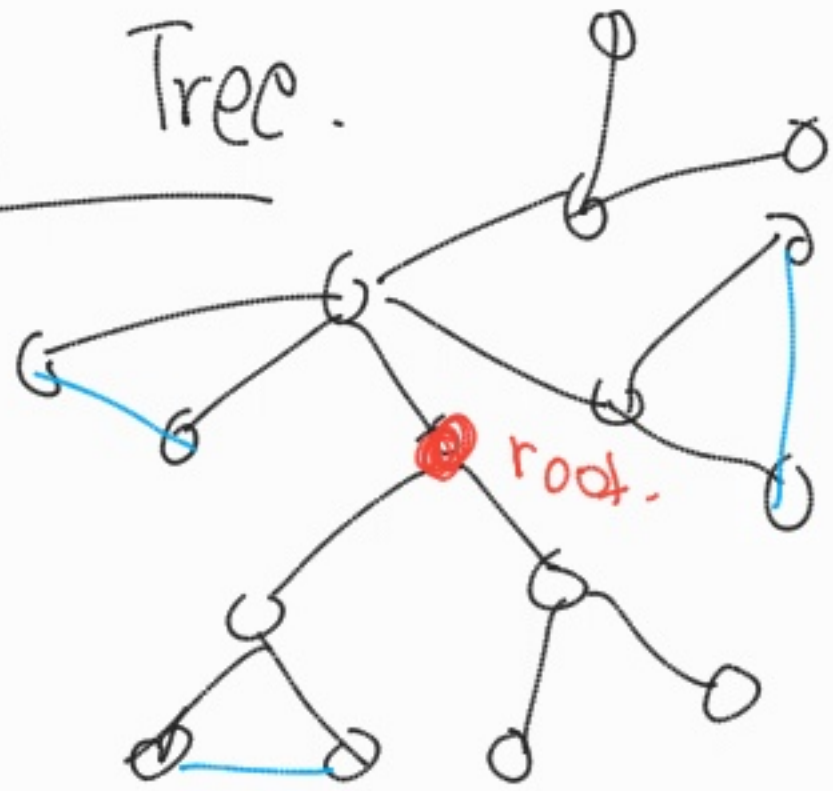
sisi : hubungan/relasi



titik : objek

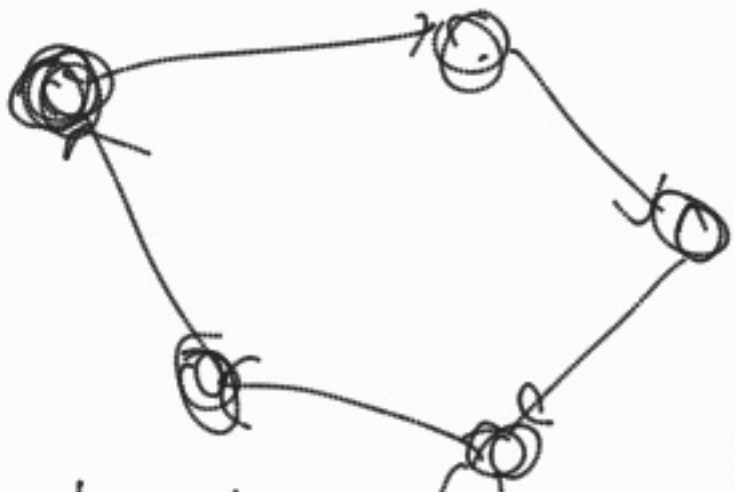
Min. Spanning Tree.

Tree :

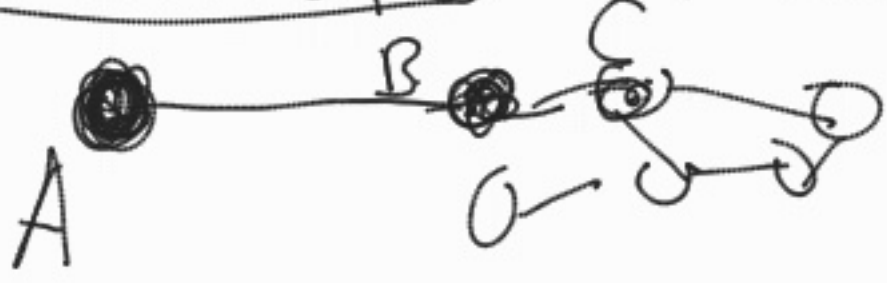


Sifat Graf pohon (Tree).

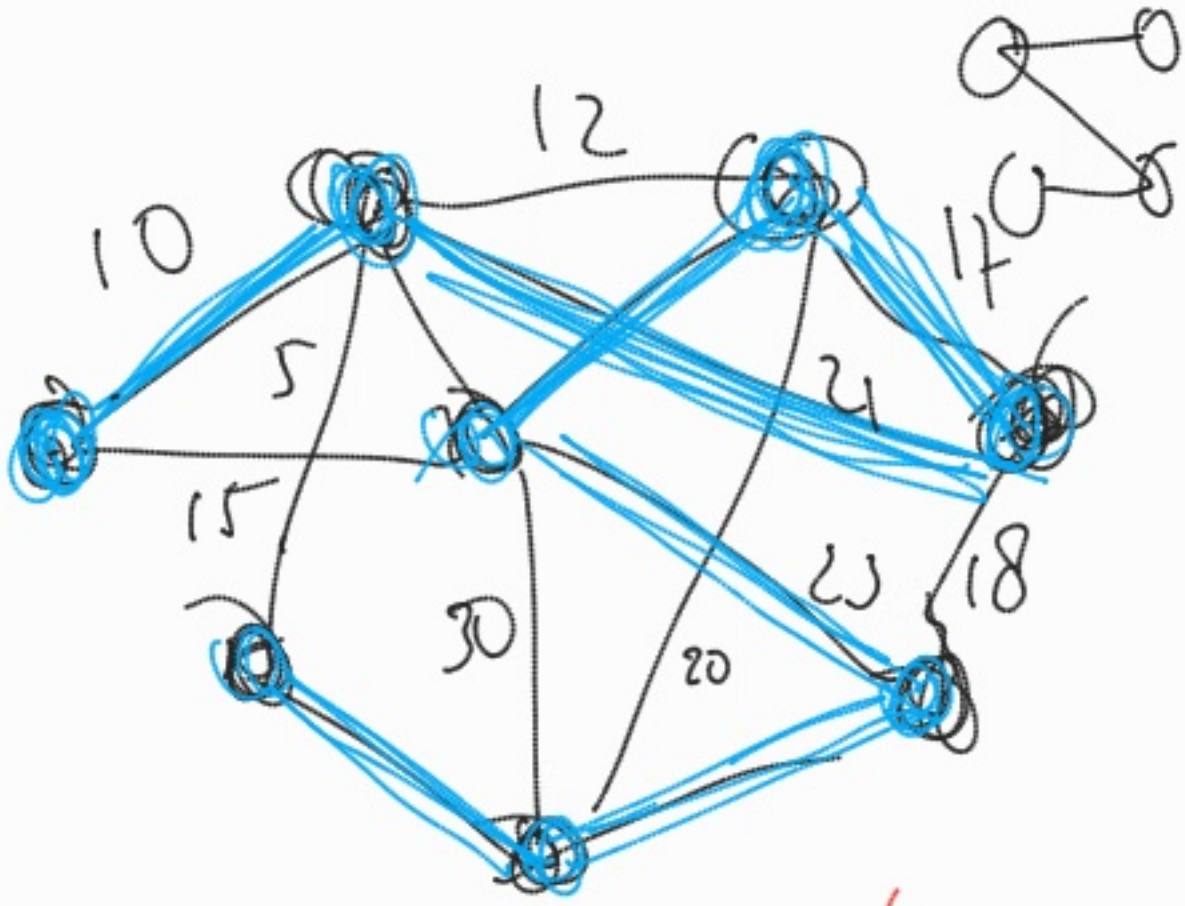
Cycle / Sirkuit.

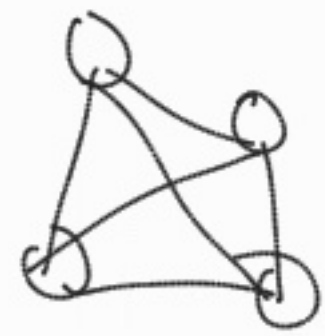


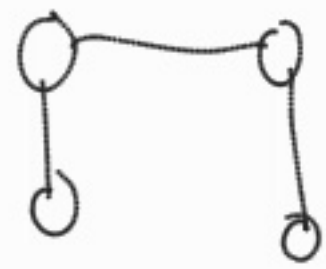
Connected / Terhubung.

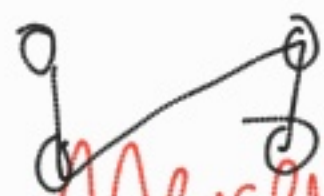


undirected.





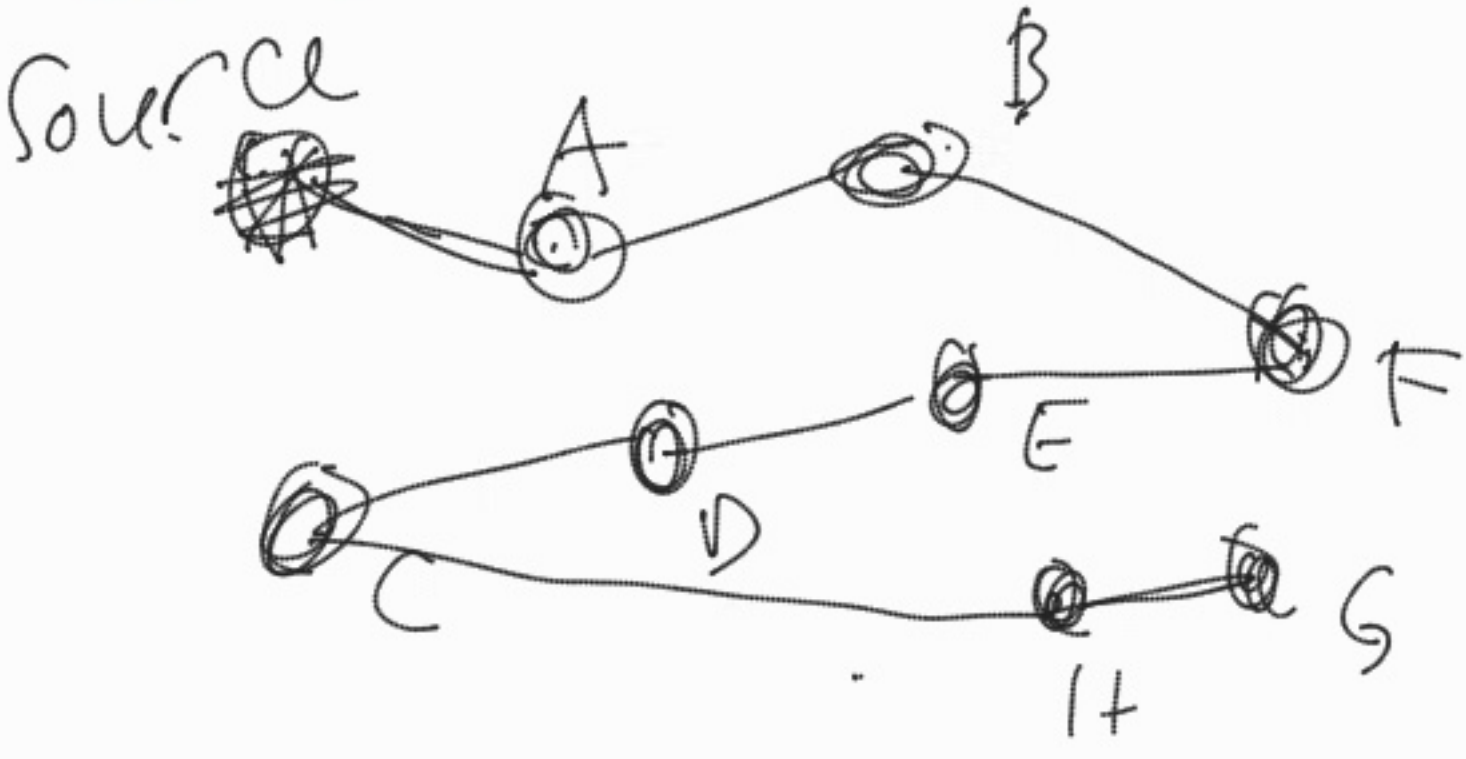




Spanning Tree / Pohon Merentang

↳ Minimum S.T.

(meminimumkan [illegible])



Semua graf terhubung

$\Rightarrow$ $\exists$ MST.

Bagaimana finding MST in a graph?

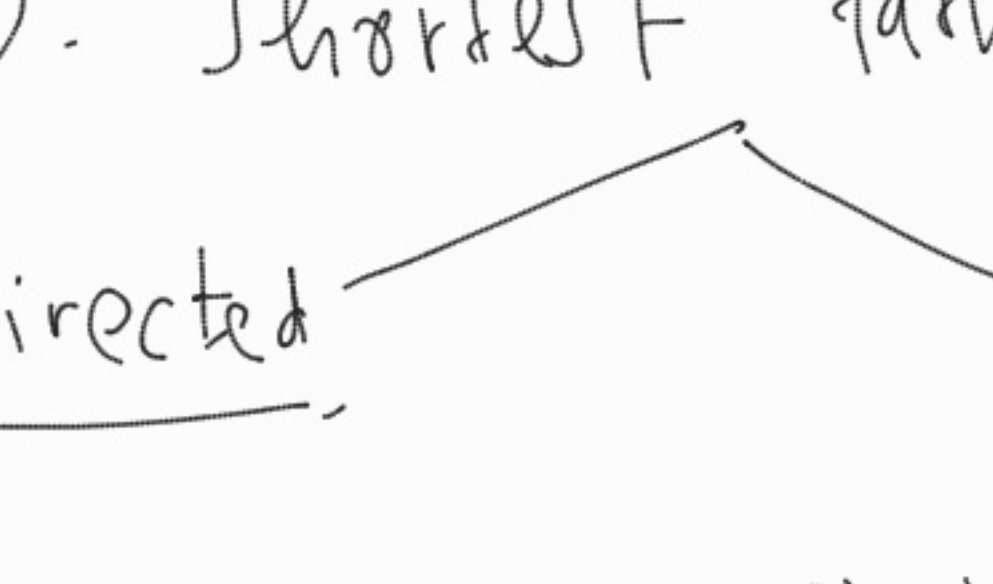
- Greedy algo. klp 2
- Kruskal algo. klp 3
- Prim algo. klp 4.

2). Shortest Path Problem

undirected — directed.

Single-source Shortest Path.

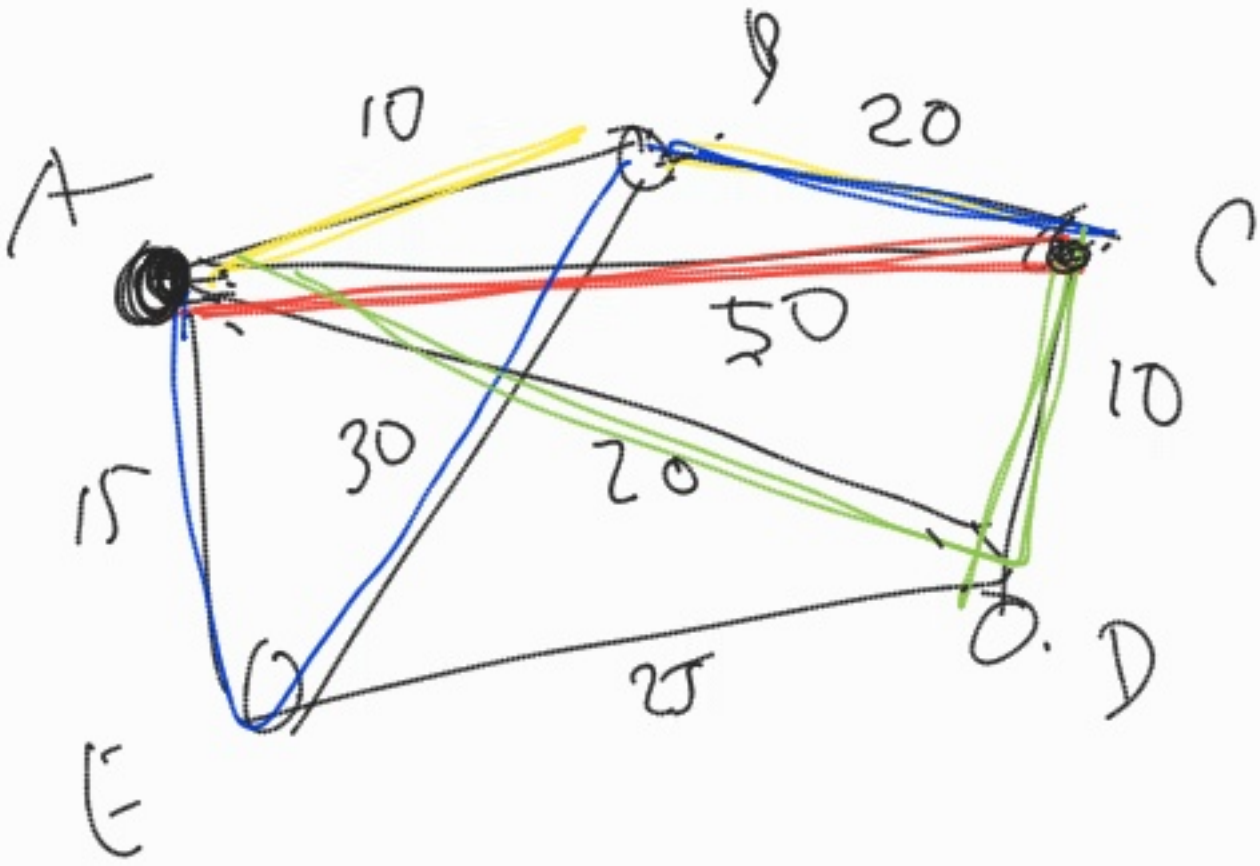
SPP : pencarian jarak min.
dari titik u ke titik v pd
graf G (∀ titik u, v).



Problem : Tentukan shortest path dari
s ke setiap titik v di G

Path lintasan

weighted/
berbobot



Path from A to C.

- AC : 50
- AB — BC : 10 + 20 = 30
- AE — EB — BC : 15 + 30 + 20 = 65
- AD — DC : 20 + 10 = 30
- dst.

Given : G dan source s.

Goal : MSP from s to
- $\forall v \in V(G)$
  $v \neq s$.

Djikstra Algo. → Flp 9,