A B , Max | A ⊆ B | ... ?

s1 = KwakKwokKwikKwekKwok
s2 = KwakKwikKwekKwokKwokKwekKwik

subsq1 = KwakKwikKwekKwok
subsq2 = KwakKwokKwok
subsq3 = KwikKwekKwok
subsq4 = KwikKwek
dst

Lcs (s1, s2) = 4

subsq3 = subsq4 + kwok
|subsq3| = 3

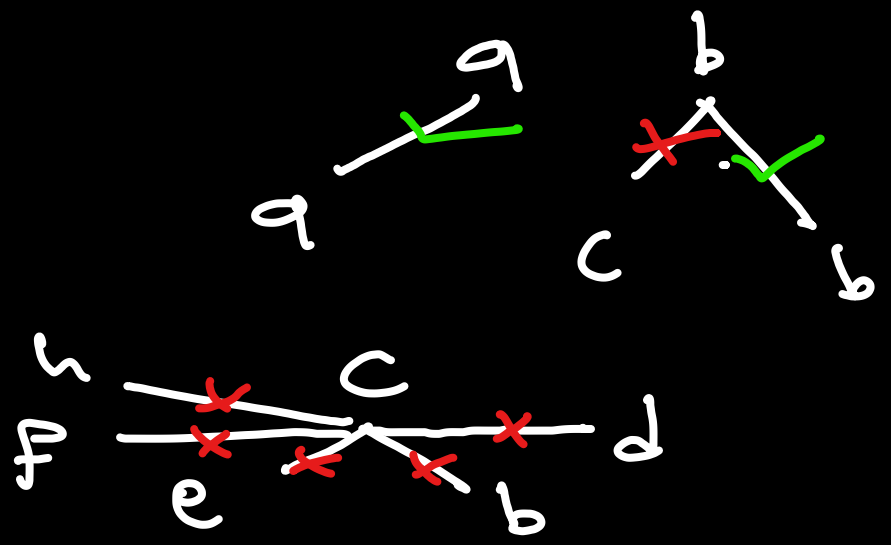
subsq2 = kwok + subsq3
|subsq3| = 4

s1 = "abcdefgh"
s2 = "acbdefh"

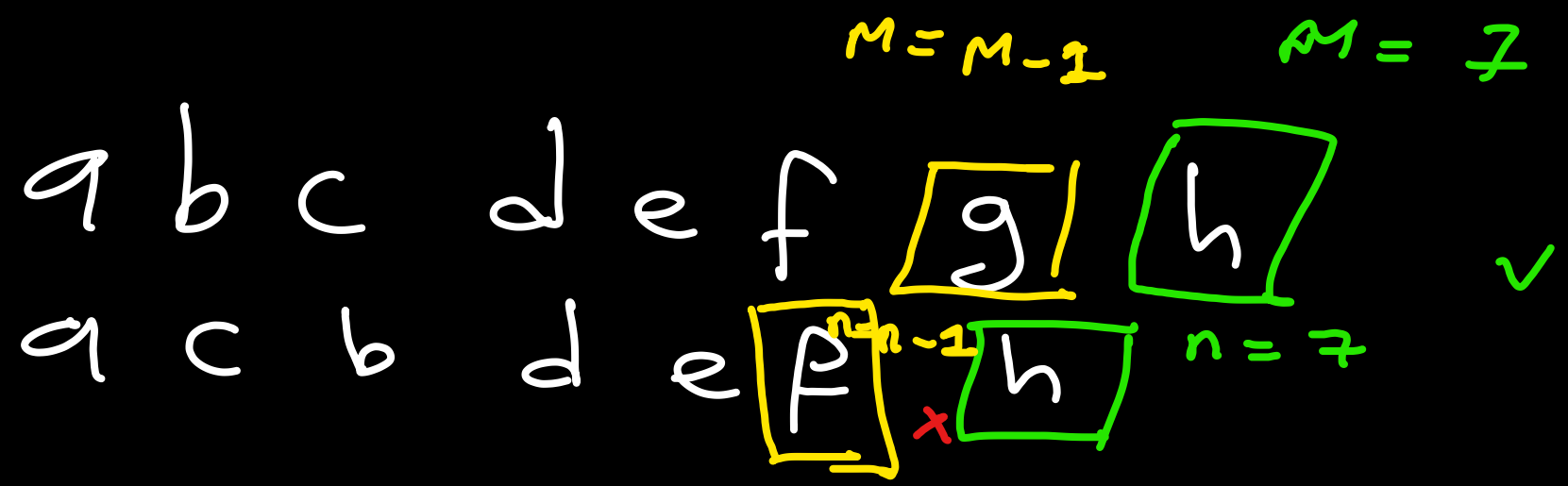
lcs(s1, s2) =

s1[i] == s2[j] → ambil

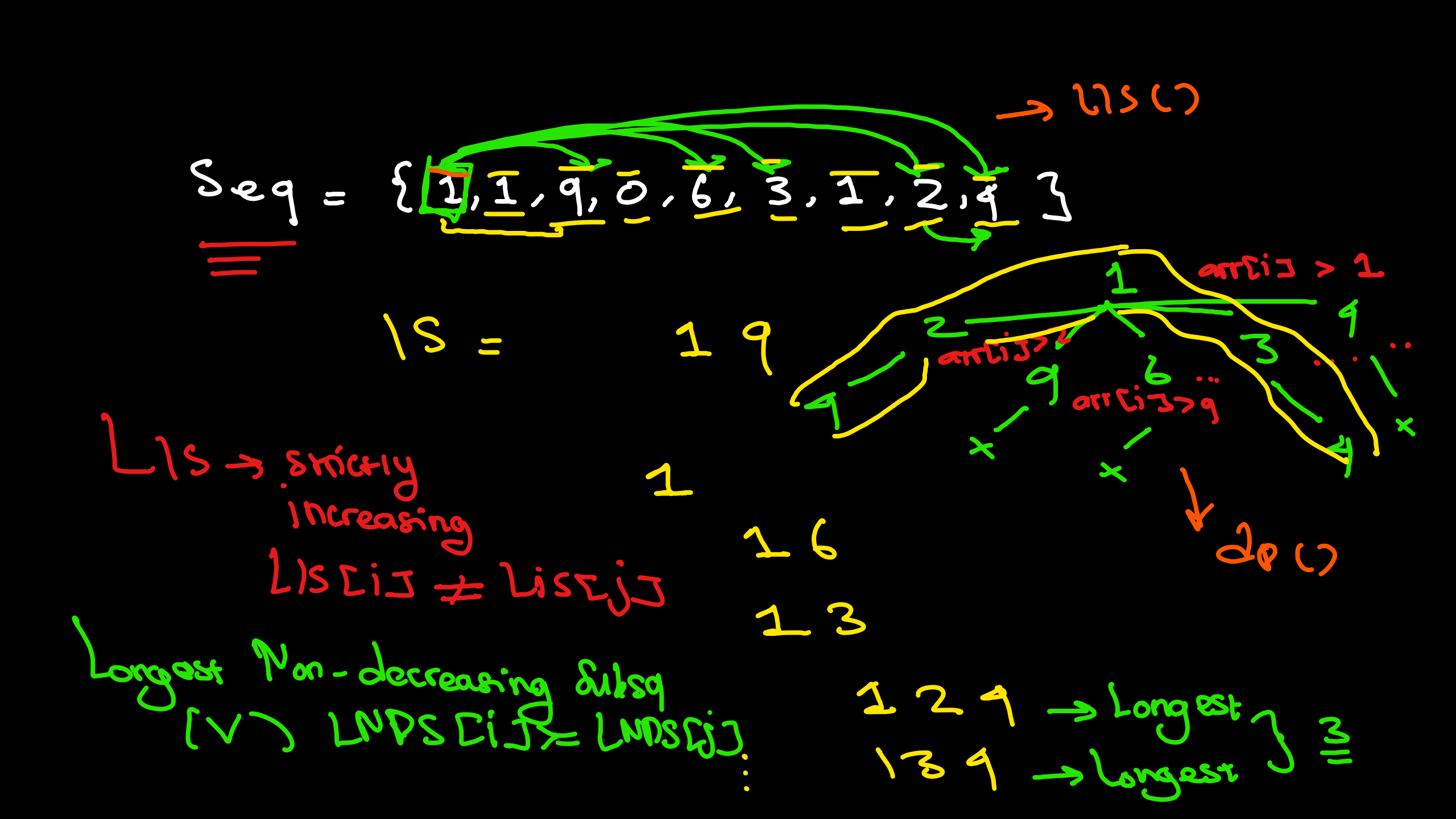
```
for(char cs1 : s1){
  for(char cs2 : s2){
        if(cs1 == cs2) ambil
            lanjut  next cs1
]
}
```




. . .




LCS → Max < M nya kita geser
N nya kita geser


Seq = { 1, 1, 9, 0, 6, 3, 1, 2, 4 }
→ LIS()
LIS = 1 9
arr[i] > 1
arr[i] > 9
arr[i] > 9
1
1 6
1 3
LIS → strictly increasing
LIS[i] ≠ LIS[j]
Longest Non-decreasing subseq
(✓) LNDS[i] ≠ LNDS[j]
1 2 9 → Longest
1 3 9 → Longest
} ≡
DP()

# 0 - 1 Knapsack Problem


N barang
barang ke - i mempunyai berat sebesar Wi, mempunyai nilai kegunaan Ci

Kita diberikan kapasitas sebuah ransel mampu menampung maksimal sebesar M kg,

Berapa nilai kegunaan maksimal yang bisa kita peroleh?

| n | Barang | Berat | Nk |
|---|---|---|---|
| 1 | Botol Minum ✓ | 5kg | 5 |
| 2 | Hp ✓ | 2kg | 7 |
| ↑ 3 | Buku | 10kg | 3 |
| ↑ 4 | Snack ✓ | 3kg | 2 |

1 / 0

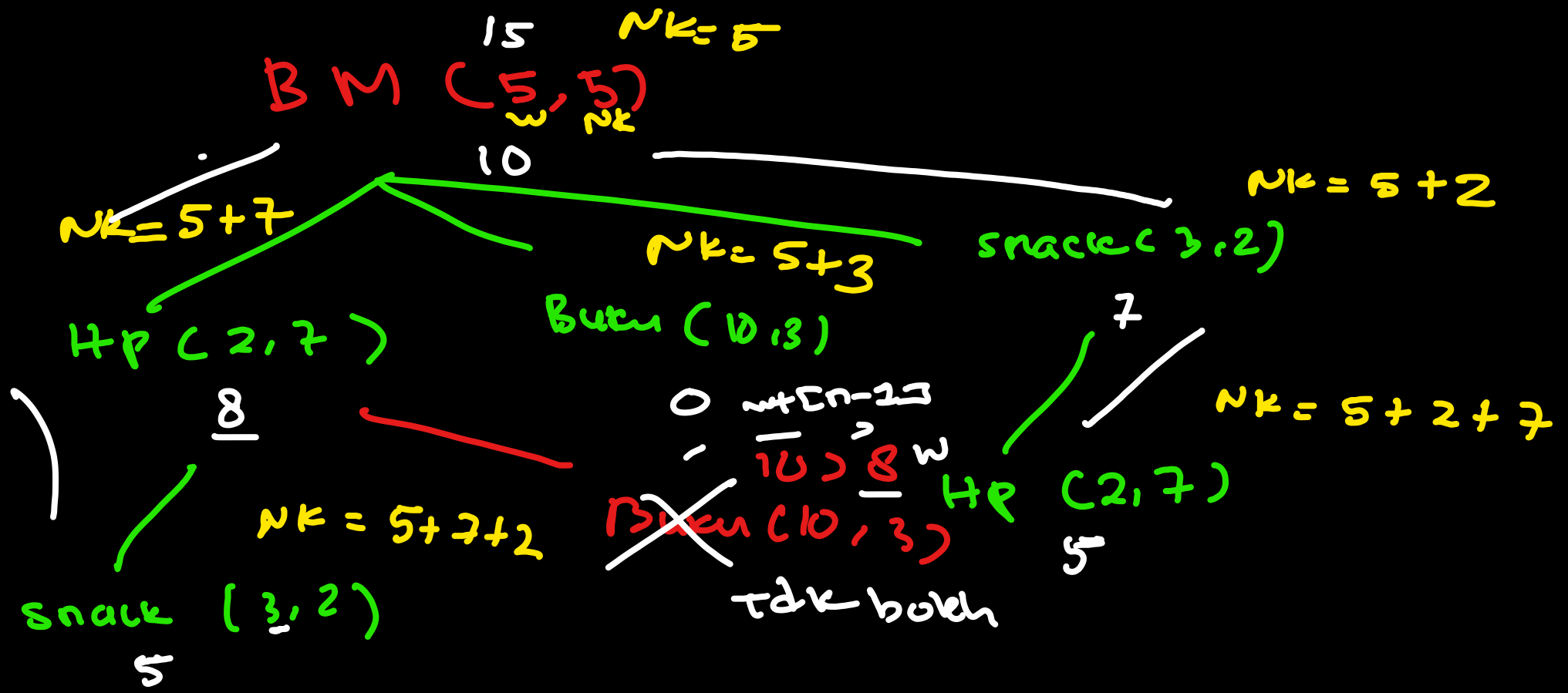
↑ Nk max = ... ?

1 / 0 14 //

1 / 0

1 / 0

* Berat Min — Nk Max
* knapsack problem


15
NK = 5
BM (5, 5)
W NK
10
NK = 5+7
NK = 5+3
NK = 5+2
Hp (2, 7)
Buku (10, 3)
snack (3, 2)
8
0
7
NK = 5+7+2
NK = 5+2+7
Hp (2, 7)
5
snack (3, 2)
5
tak boleh