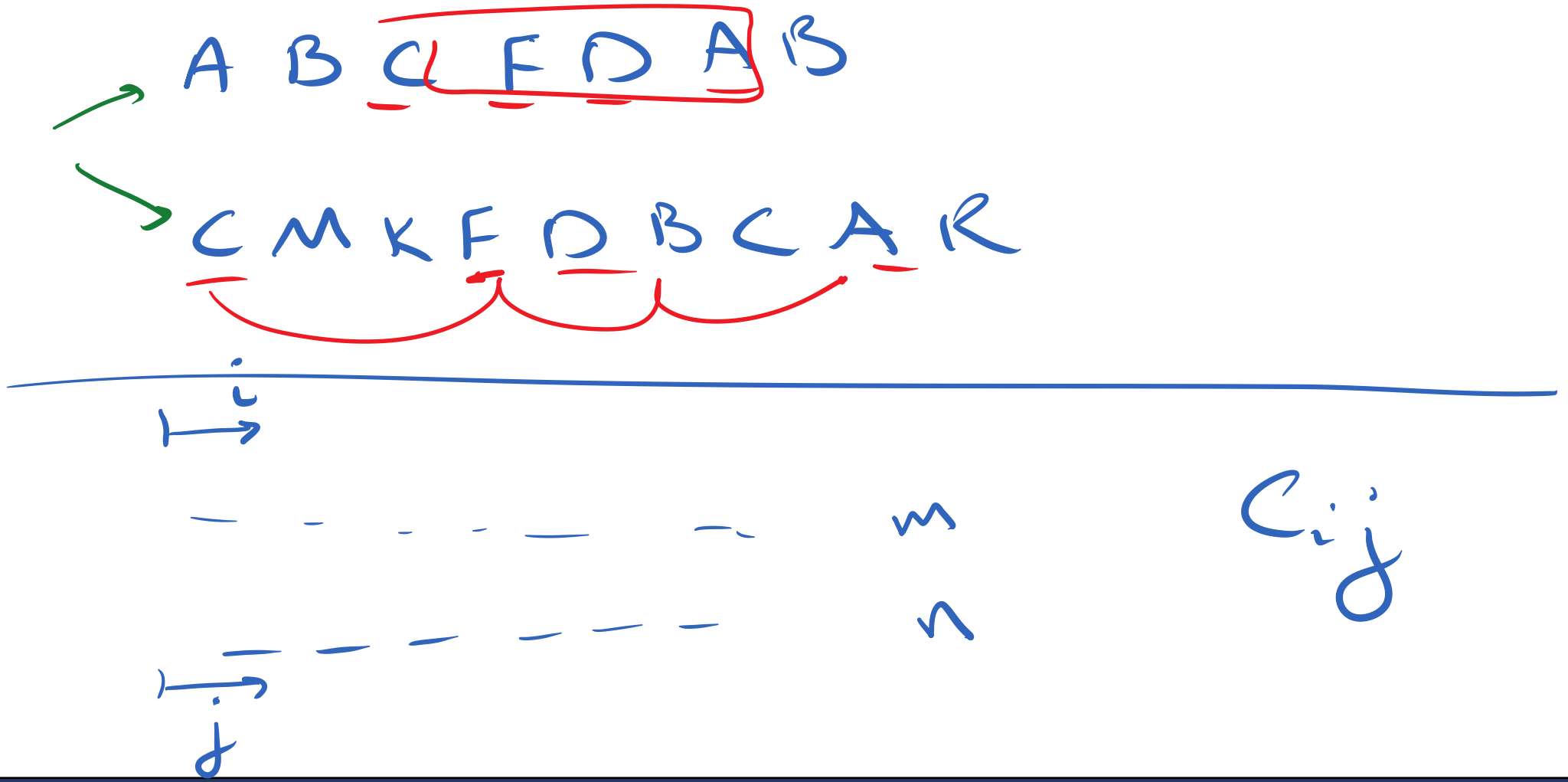
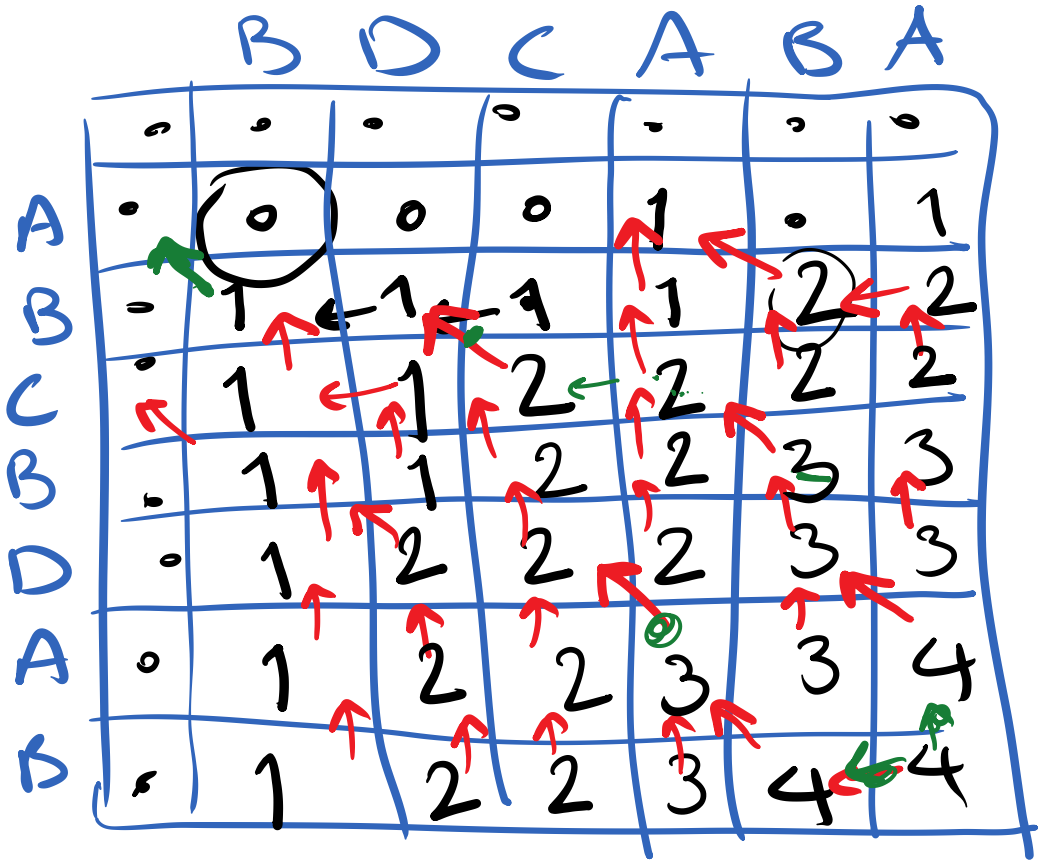
LCS

بزرگترین زیردنباله مشترک



Seq1: B D C A B A

Seq2 : A B C B D A B



BDAB

BCBA

BCAB

c_{ij}

$$c_{ij} = \begin{cases} c_{i-1,j-1}+1 \\ \max(c_{i-1,j}, c_{i,j-1}) \end{cases}$$

```
def LCS (A, B):
    n = A.length
    m = B.length
    C = Zeros(n+1, m+1)
    for i = 1 → n
        for j = 1 → m
            if A[i] = B[j] → c[i-1][j-1] + 1
            elseif c[i-1][j] > c[i][j-1] → c[i-1][j]
            else c[i-1][j] < c[i][j-1] → c[i][j-1]
    c[i][j] ←
```

زمان اجرا $O(nm)$

Greedy

حریصانه

کوله پشتی

def fractional knapsack (weights, prices, W)

$k = 0$; $i = 1$; $p = 0$

weights ← Quick-Sort (prices / weights)

prices

Greedy

$X = \text{Zeros}(1, n)$

while $(k != W)$:

{ if $(k + w[i] \leq W)$

$x[i] = 1$; $p = p + p[i]$

else :

$O(n \log n)$

$$u[i] = \frac{W - k}{w[i]} \quad , \quad p = p + \frac{w - k}{w[i]} \times p[i]$$

i++

}

return $\underset{\sim}{X}$, $\underset{\sim}{P}$