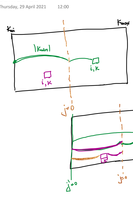
Sem11-B3

Thursday, 29 April 2021 12:00



$K_{min} = \min_{\substack{i \le k \le n \\ j=0 \\ x[i]=y[j]}} k$

$K_{max} = \max_{\substack{i \le k \le n \\ j=0 \\ x[i]=y[j]}} k$

dacă $K_{min} < 0 \rightarrow add = |K_{min}|$

$k' = k + |K_{min}|$

add

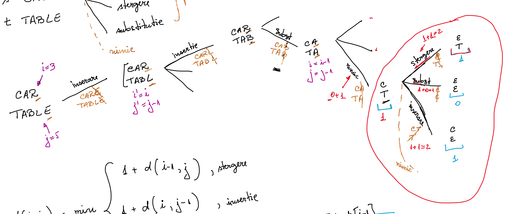
add pt $y[j]$ se găsește la $x[i][K_{min}+k]$

dacă $K_{max} \geq 0 \rightarrow add = 0$

---

s CAR

t TABLE



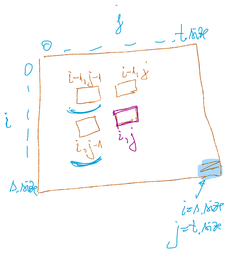
$$d(i,j) = \min \begin{cases} 1 + d(i-1, j) & \text{, ștergere} \\ 1 + d(i, j-1) & \text{, inserție} \\ \cancel{1 + d(i-1, j-1)} & \text{, substituție dacă } s[i-1] \neq t[j-1] \\ d(i-1, j-1) & \text{, nimic dacă } s[i-1] = t[j-1] \end{cases}$$

2a) ștergere și inserție

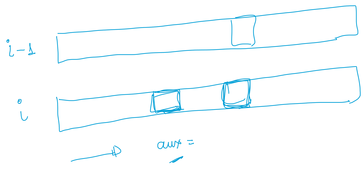
$d[i][j]$ - salvez rezultatul pt $dist(i,j)$

└ inițial -1

```
dist(i, j) modifies s[], t[]
{   if(d[i][j] != -1) return d[i][j];
    if(i==0 || j==0){ d[i][j]=i+j; return i+j; }
    m = min( 1+dist(i-1, j) , 1+dist(i, j-1) );
    if(s[i-1]==t[j-1])
        m = min(m, dist(i-1, j-1));
    d[i][j]=m;
    return m;
}
```




```
for(i=0; i<=s.length; i++)
for(j=0; j<=s.length; j++)
{ * dist(..) -> d(...)
```



Ex 2b

$v = [2, 5, 1, 3, 6, 9, 4, 3]$ $n = 7$

$d = [1, 2, 1, 2, 3, 4, 3]$ ← predecesorul (după care subșir am adăugat)

$p = [-1, 0, -1, 0, 1, 4, 3]$

- lungimea celui mai lung subșir crescător

└ nu neapărat el consecutive în v

2
25
1
23
256
2569
234

$d(i)$ = lungimea celui mai lung subșir care se termină cu $v[i]$

$$d(i) = 1 + \max_{\substack{0 \le j < i \\ v[j] \le v[i]}} d(j)$$

dacă nu există, $\max = 0$

```
res = 0   // poziția rezultatului
for(i=0; i<n; i++)
{   p[i] = -1;
    max = 0;
    for(j=0; j<i; j++)
        if(v[j] <= v[i] && d[j] > max)
            { max = d[j]; p[i] = j; }
    d[i] = 1 + max;
    if(d[res] < d[i]) res = i;
}
return d[res];
```

// print $v[\underbrace{res}_{k}]$; $v[\underbrace{p[\overbrace{res}^{k}]}]$, $v[p[p[res]]]$ ....

```
k = res;
do {  print v[k];
      k = p[k]
} while(k != -1);   // afișează în ordine inversă
```

Temă: afișare în ordine crescătoare

+ Ex 2c, 2d