http://mj.ucw.cz/vyuka/ads1/

Průvodce labyrintem alg.

Úloha: Je dána posl. $x_1 — x_n \in \mathbb{Q}$.
Najděte nejdelší rostoucí podposloupnost.

① Binární počítadlo $b_1 — b_n$

Prostor: $\Theta(n)$

Krok: $i \leftarrow 1$
Dokud $b_i = 1$:
$b_i \leftarrow 0$
$i \leftarrow i+1$
$b_i \leftarrow 1$

$\} O(n)$

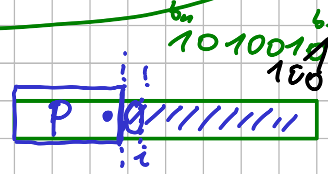
Čas: 2^n podposl.
celkem $\Theta(n \cdot 2^n)$

Zastavení: $b_{n+1} \leftarrow 0$
až se změní na 1, konec

kontrola: $p \leftarrow -\infty$ předchůdce
$d \leftarrow 0$ akt. délka
Pro $j = 1, \dots, n$:
Pokud $b_j = 1$:
Pokud $x_j > p$:
$d \leftarrow d+1$
$p \leftarrow x_j$
Jinak:
vrátíme se
nejdelší $\leftarrow$ max(nejdelší, d)

$\Theta(n)$

② Rekurze



gen(P, i):
Pokud $i > n$: zpracujeme P

gen$(P, i+1)$
gen$(P \| x_i, i+1)$

init: gen$(\emptyset, 1)$

②* Vylepšená rekurze

$g(d, p, i)$:
délka RPP — posl. prvek RPP — kde jsem ve vstupu

Pokud $i > n$: nejdelší $\leftarrow$ max(nejdelší, d)
vrátíme se

$g(d, p, i+1)$
Pokud $x_i > p$:
$g(d+1, x_i, i+1)$

init: $g(0, -\infty, 1)$

③ Rekurze pozpátku

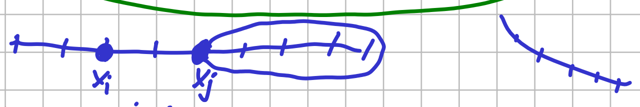
init: přidáme $x_0 \leftarrow -\infty$
$f(0) - 1$

$f(i)$:= max. délka RPP začínající prvkem x_i

$0 \dots n$

$d \leftarrow 1$
Pro $j = i+1, \dots, n$:
Je-li $x_j > x_i$:
$d \leftarrow \max(d, f(j)+1)$
Vrátíme d

☹ stále exponenciální



$i < j$
$x_i < x_j$

③* Přidáme kešování

pamatujeme si, jaké $f(i)$ už jsme spočítali
& nepočítáme znovu

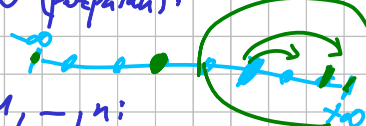
$\Rightarrow$ provedeme max. $\underbrace{n+1}_{O(n)}$ volání funkce f
každé volání trvá $O(n)$

celkem čas $O(n^2)$

④ Přímý výpočet tabulky z ③* pozpátku

nechť $f_i := f(i)$

Pro $i = n, \dots, 0$ (pozpátku):
$f_i \leftarrow 1$
Pro $j = i+1, \dots, n$:
Je-li $x_j > x_i$:
$f_i \leftarrow \max(f_i, f_j + 1)$
Vrátíme $f_0 - 1$



Čas $\Theta(n^2)$
dynamické programování

⑤ Grafová úloha

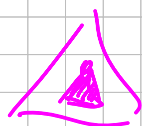
Vrcholy $V := \{1 — n\}$
Hrany (šipky) $(i,j) \in E \equiv i < j \;\&\; x_i < x_j$

cesty v grafu $\Leftrightarrow$ RPP
$\rightarrow$ hledáme nejdelší cestu

- přidáme $x_0 \leftarrow -\infty$, $x_{n+1} \leftarrow +\infty$
  $\Rightarrow$ hledáme cestu z $-\infty$ do $+\infty$
- nejdelší cesta obecně těžká,
  ale náš graf je acyklický
  $\Rightarrow \exists$ alg. se složitostí
  $\Theta(|V| + |E|) = \Theta(n^2)$

⑥ Datová struktura pro ④

↳ pamatuje si množinu dvojic (x_j, f_j) — klíč: x_j, hodnota: f_j



Pro $i = n, \dots, 0$:
$f \leftarrow \max\left(1, \text{RangeMax}(x_{i+1}, +\infty) + 1\right)$
Vložíme (x_i, f) do DS.

Umí: Vložit dvojici

Najít max. z hodnot pro klíč ∈ interval

$\text{RangeMax}(s,t) := \max\{ f_j \mid (x_j, f_j) \text{ je dvojice } \& \ x_j \in [s,t] \}$

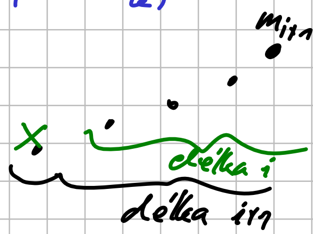
Lze implementovat v čase $O(\log n)$. → celkem $O(n)$ operací → čas $O(n \log n)$

---

⑦ Konečné řešení

- postupně přidáváme $x_1, x_2, \dots, x_n$ (zatím do x_k)
- udržujeme $m_i :=$ min. koncový prvek RPP délky i
- co se stane, když přidáme x_{k+1}
  ↳ jaké RPP vzniknou?

$m_1 = \min(x_1 \dots x_k)$
$m_0 := -\infty$
$m_i \le m_{i+1}$



$-\infty \le \le \le \le \le +\infty \dots\dots +\infty$
0 1 - - -
$< x_{k+1}$ | $\ge x_{k+1}$ | tady RPP už ∄

přepíšeme na x_{k+1}

rozhraní najdeme binárně → čas $O(\log n)$

za každou RPP nějaké délky i končící $< x_{k+1}$ — přidáme x_{k+1} → RPP délky $i+1$ s koncem x_{k+1}

jde, pokud $m_i < x_{k+1}$

pokud $m_{i+1} > x_{k+1}$,
pak $m_{i+1} \leftarrow x_{k+1}$

a zbytek $O(1)$
↳ čas celkem $O(n \log n)$

Random Access Machine (RAM) — výpočetní model (abstraktní stroj)

- počítá s čísly … celými
  - vše ostatní kódujeme do čísel
- paměť: pole čísel indexované čísly
  ↳ buňky paměti

{ | | … | 1 | 3 | -7 | 3 | 42 | … | | }
… -2 -1 0 1 2 …

- instrukce:
  - přiřazení kam ← co
  - aritmetika $z \leftarrow x + y$

operandy:
zdroj: literál (číslo) — 42
buňka paměti — [42]
cíl: nepřímá adresace — [[42]] (buňka, jejíž adresa je v buňce 42)

Př.: [1] ← [2] + 7
[1] ← [x] − 1
[[−25]] − 1

konvence: lokální proměnné
A := [−1]
B := [−2]
⋮
Z := [−26]

operace: + − * / mod
bitový & | ^
AND OR XOR
<< >> bit. posun
} jen pro nezáporná čísla!

7 & 14 = 6
… 0111
… 1110
0110