

# Algoritmi in podatkovne strukture 2 — prvi izpitni rok

## 19. junij 2026

Naloge so enakovredne, podnaloge pa ne nujno. Za reševanje imate na voljo 100 minut časa.

Vse odgovore obvezno utemeljite!

- ① »Kako lahko APS predava nekdo, ki še nikoli ni napisal svojega algoritma za urejanje?« sem ondan ujel na hodniku. Prmejduš, da imajo prav, pomislim. Pa poskusimo:

```
▷ Uredi  $A[1 : n]$ . ◁  
function FÜRSTSORT( $A, n$ )  
  if  $n \geq 2$  then  
     $r \leftarrow \lfloor 2n/3 \rfloor$   
    FÜRSTSORT( $A, r$ )  
    for  $i \leftarrow r + 1$  to  $n$  do  
      ▷ Vstavi  $A[i]$  na ustrezno mesto levo od  $i$ . ◁  
       $x \leftarrow A[i]$   
       $j \leftarrow i - 1$   
      while  $j \geq 1 \wedge A[j] > x$  do  
         $A[j + 1] \leftarrow A[j]$   
         $j \leftarrow j - 1$   
       $A[j + 1] \leftarrow x$   
  *****
```

- (a) Kakšen izpis bomo dobili pri izvedbi funkcije FÜRSTSORT na tabeli  $A = [8, 7, 6, 5, 4, 3, 2, 1]$  in pri  $n = 8$ , če na mesto, označeno z \*\*\*\*\* (torej po zaključku zanke for), postavimo klic funkcije za izpis celotne vsebine tabele  $A$ ?
- (b) Zapišite rekurenčno enačbo za določitev časovne zahtevnosti funkcije FÜRSTSORT ( $T(n) = \dots$ ).
- (c) Določite časovno zahtevnost funkcije FÜRSTSORT.
- (d) S substitucijsko metodo dokažite, da je vaša ocena iz prejšnje točke pravilna. Zadošča, če pravilnost dokažete za zgornjo mejo ( $T(n) = O(\cdot)$ ).

### Rešitve

- (a)  $[7, 8, 6, 5, 4, 3, 2, 1]$   
 $[6, 7, 8, 5, 4, 3, 2, 1]$   
 $[4, 5, 6, 7, 8, 3, 2, 1]$   
 $[1, 2, 3, 4, 5, 6, 7, 8]$
- (b)  $T(n) = T(2n/3) + \Theta(n^2)$ .
- (c) Po krovnem izreku je  $a = 1$ ,  $b = 3/2$  in  $d = 2$ . Ker je  $a < b^d$ , dobimo  $T(n) = \Theta(n^d) = \Theta(n^2)$ .
- (d) Postavimo trditev  $T(n) \leq cn^2$  za vsak  $n \geq n_0$  in nek  $c > 0$ . Po induktivnem sklepanju predpostavimo, da trditev velja za vsak  $n' < n$ , sedaj pa preverimo, ali velja tudi za  $n$ :

$$T(n) = T(2n/3) + \Theta(n^2)$$

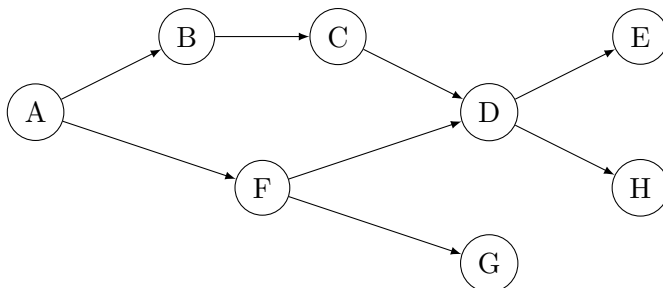
$$\begin{aligned} &\leq c(2n/3)^2 + \Theta(n^2) \\ &\leq 4cn^2/9 + kn^2 \\ &= (4c/9 + k)n^2 \end{aligned}$$

Konstanto  $c$  lahko poljubno izberemo. Če jo izberemo tako, da velja  $k \leq 5c/9$  (torej  $c \geq 9k/5$ ), dobimo

$$\begin{aligned} T(n) &\leq (4c/9 + 5c/9)n^2 \\ &= cn^2 \end{aligned}$$

Trditev je tako dokazana.

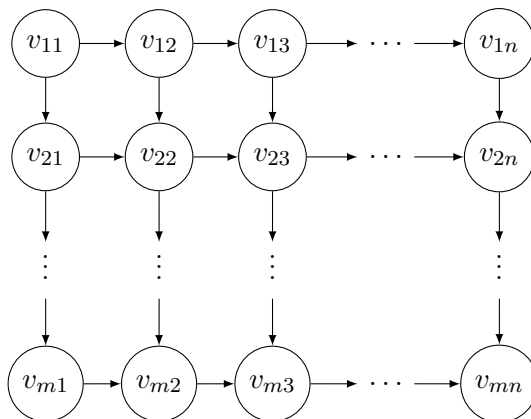
- ② Podan je usmerjen acikličen graf (DAG)  $G = (V, E)$ . Naj bo *dokončna pot iz vozlišča  $v$*  pot, ki se prične v vozlišču  $v$  in je ni mogoče podaljšati. Na primer, sledeči graf ima 5 dokončnih poti iz vozlišča A (ABCDE, ABCDH, AFDE, AFDH in AFG), iz vozlišča G pa imamo le eno takšno pot (G):



Naj bo  $D(v)$  število dokončnih poti iz vozlišča  $v$ .

- (a) Dokažite, da za usmerjeni mrežni graf  $m \times n$  (spodnja slika) pri  $m \geq 1$  in  $n \geq 1$  velja

$$D(v_{11}) = \binom{m+n-2}{m-1}.$$



Namig: popolna indukcija, kaj pa drugega!

- (b) Dobil sem takšno idejo za izračun  $D(v)$  za določeno vozlišče  $v$  v podanem DAG: Nastavi globalni števec  $D$  na 0, nato pa poženi DFS iz vozlišča  $v$ . Vsakokrat, ko naletiš na vozlišče brez sosedov, povečaj  $D$  za 1. Ko se DFS konča, nastavi  $D(v) \leftarrow D$ .

Dokažite, da ta algoritem deluje za poljuben DAG, ali pa poiščite protiprimer.

- (c) Za podani DAG z  $n$  vozlišči in  $O(n)$  povezavami bi radi izračunali  $D(v)$  za *vsako* vozlišče  $v$ . Opišite algoritem, ki to doseže v času  $O(n)$ .

## Rešitve

- (a) Naj bo  $P(m, n) \equiv d(v_{00})$  pri grafu  $m \times n$ . Če je  $m = n = 1$ , je število dokončnih poti iz edinega vozlišča enako 1, to pa je hkrati  $\binom{0}{0}$ . Sedaj pa predpostavimo, da trditev velja za vse pare  $(m', n')$ , kjer je  $m' < n$  in  $n' \leq n$  ali  $m' \leq m$  in  $n' < n$ , in preverimo, ali velja tudi za par  $(m, n)$ . Pri tem upoštevamo, da je število dokončnih poti iz vozlišča  $v_{ij}$  enako vsoti števila dokončnih poti iz njegovega desnega sosedu in števila dokončnih poti iz njegovega spodnjega sosedu. To pomeni, da velja

$$P(m, n) = P(m-1, n) + P(m, n-1),$$

saj v grafu ne moremo potovati niti v levo niti navzgor. Z upoštevanjem induktivne predpostavke in znane formule

$$\binom{p}{q} = \binom{p-1}{q-1} + \binom{p-1}{q}$$

dobimo

$$\begin{aligned} P(m, n) &= P(m-1, n) + P(m, n-1) \\ &= \binom{m+n-3}{m-2} + \binom{m+n-3}{m-1} \\ &= \binom{m+n-2}{m-1} \end{aligned}$$

in trditev je dokazana.

- (b) Ta algoritem seveda ni pravilen, saj odpove vsakokrat, ko je do istega vozlišča mogoče priti po več poteh. Najenostavnejši protiprimer je graf s povezavami (1, 2), (2, 3) in (1, 3). Algoritem bi  $D(1)$  izračunal kot 1, v resnici pa je rezultat enak 2.
- (c) Če vozlišča topološko uredimo, potrebujemo za to  $\Theta(n + O(n)) = \Theta(n)$  časa. Sedaj lahko posamične  $D(v)$  izračunamo po sledečem algoritmu:

**function** IZRAČUNAJD( $G$ )

Naj bodo  $v_1, \dots, v_n$  vozlišča v *obratnem* topološkem vrstnem redu

**for**  $i \leftarrow 1$  **to**  $n$  **do**

▷  $\mathcal{N}(v)$  je množica sosedov vozlišča  $v$  ◁

**if**  $\mathcal{N}(v_i) = \emptyset$  **then**

$D(v_i) \leftarrow 1$

**else**

$D(v_i) \leftarrow 0$

**for**  $w \in \mathcal{N}(v_i)$  **do**

$D(v_i) \leftarrow D(v_i) + D(w)$

Če so vozlišča urejena v obratnem topološkem vrstnem redu, potem so pri računanju  $D(v)$  vrednosti  $D(\cdot)$  za sosedo vozlišča  $v$  že znane. Kljub dvojni zanki je časovna zahtevnost algoritma  $\Theta(n)$ , saj algoritem po zaključku topološkega urejanja po enkrat obravnava vsako vozlišče in vsako povezavo.

- ③ Naj bo  $D(S)$  število medsebojno različnih nepraznih podnizov niza  $S$ . Na primer, niz **ababa** ima 9 takih podnizov: **a**, **b**, **ab**, **ba**, **aba**, **bab**, **abab**, **baba** in **ababa**. Vsak podniz štejemo samo po enkrat, tudi če v vhodnem besedilu nastopa po večkrat.

Naj bo  $SA$  priponska tabela,  $LCP$  pa tabela najdaljših skupnih predpon pripon niza  $S$ .

- (a) Zapišite tabeli  $SA$  in  $LCP$  za niz **ababa**.
- (b) Vsak niz  $S$  vsebuje  $|S|$  medsebojno različnih nepraznih *predpon* (in seveda tudi pripon), niza  $S$  in  $T$  pa lahko vsebujeta manj kot  $|S| + |T|$  različnih predpon, ker se lahko nekatere podvajajo. Koliko medsebojno različnih nepraznih predpon vsebujeta priponi na mestih  $i - 1$  in  $i$  v leksikografskem vrstnem redu pripon poljubnega niza  $S$  dolžine  $n$ ? (V odgovoru boste najbrž uporabili tabeli  $SA$  in  $LCP$ .)
- (c) Zapišite algoritem, ki za poljubno besedilo  $T$  v času  $O(n)$  izpiše število medsebojno različnih nepraznih podnizov, če predpostavimo, da sta tabeli  $SA$  in  $LCP$  za besedilo  $T$  že izdelani. (Namig: zadošča en sam sprehod po priponah niza.)

### Rešitve

- (a) Izhajamo iz zaporedja leksikografsko urejenih pripon (prvi stolpec).

$T[SA[i] :]$	$SA[i]$	$LCP[i]$
<b>a</b>	5	0
<b>aba</b>	3	1
<b>ababa</b>	1	3
<b>ba</b>	4	0
<b>baba</b>	2	2

- (b) Pripona, ki se nahaja na mestu  $i - 1$  v leksikografskem zaporedju, je dolga  $n - SA[i - 1] + 1$ , zato vsebuje natanko toliko nepraznih predpon. Pripona na mestu  $i$  seveda vsebuje  $n - SA[i] + 1$  nepraznih predpon. Upoštevati pa moramo, da je  $LCP[i]$  takih predpon skupnih. Število medsebojno različnih predpon omenjenih pripon je potemtakem

$$2n + 2 - SA[i - 1] - SA[i] - LCP[i].$$

- (c) Po priponah niza se sprehodimo v leksikografskem vrstnem redu. Če je tabela  $SA$  že izdelana, je ta vrstni red enostavno  $T[SA[1] :]$ ,  $T[SA[2] :]$ ,  $\dots$ ,  $T[SA[n] :]$ . Pri sprehodu upoštevamo, da  $i$ -ta pripona dolžine  $k = n - SA[i] + 1$  predstavlja  $k$  podnizov (vsaka predpona te pripone je eden od podnizov niza  $S$ ), vendar pa je samo  $k - LCP[i]$  od teh podnizov takih, da niso vključeni v predhodno obiskane pripone. Na primer, pri nizu **ababa** imamo takšno situacijo (podnizi, ki jih predstavljajo predhodno obiskane pripone in jih zato ne smemo več šteti, so prečrtani):

$T[SA[i] :]$	$LCP[i]$	Podnizi
<b>a</b>	0	<b>a</b>
<b>aba</b>	1	<del>a</del> , <b>ab</b> , <b>aba</b>
<b>ababa</b>	3	<del>a</del> , <del>ab</del> , <del>aba</del> , <b>abab</b> , <b>ababa</b>
<b>ba</b>	0	<b>b</b> , <b>ba</b>
<b>baba</b>	2	<del>b</del> , <del>ba</del> , <b>bab</b> , <b>baba</b>

Algoritem je sedaj enostaven:

```
function PODNIZI( $S$ ,  $SA$ ,  $LCP$ )  
   $n \leftarrow |S|$   
   $D \leftarrow 0$   
  for  $i \leftarrow 1$  to  $n$  do  
     $D \leftarrow D + n - SA[i] + 1 - LCP[i]$   
  return  $D$ 
```

- ④ Podana je šahovnica  $N \times N$  (v obeh dimenzijah gredo koordinate od 1 do  $N$ ), figure pa naj imajo številke od 1 do  $F$ . Pri običajnem šahu imamo  $N = 8$  in  $F = 6$  (npr. 1 = kralj, 2 = dama, 3 = trdnjava, 4 = lovec, 5 = skakač, 6 = kmet). Barvo figure zanemarimo. Figura  $f$  na polju  $(x, y)$  naj bo predstavljena s točko  $(x, y, f)$ .

Za podano šahovnico z razmeščenimi figurami bi radi zgradili KD-drevo, pri katerem na nivojih 0, 3, 6, ... delimo množico točk po koordinati  $x$ , na nivojih 1, 4, 7, ... po koordinati  $y$ , na nivojih 2, 5, 8, ... pa po številki figure.

- (a) Narišite KD-drevo za točke  $A(1, 8, 5)$ ,  $B(2, 7, 4)$ ,  $C(3, 6, 3)$ ,  $D(4, 5, 2)$  in  $E(5, 4, 1)$ . Če je več enakovrednih možnosti, lahko izberete katerokoli od njih.
- (b) Če množica točk vsebuje točki  $T(x, y, f)$  in  $T'(x', y', f')$  z lastnostjo  $x = x' \vee y = y' \vee f = f'$ , lahko v določenih okoliščinah nastane težava. Kakšne so te okoliščine in kako bi rešili nastalo težavo?
- (c) Sledeča slika prikazuje začetno razporeditev figur pri običajnem šahu:

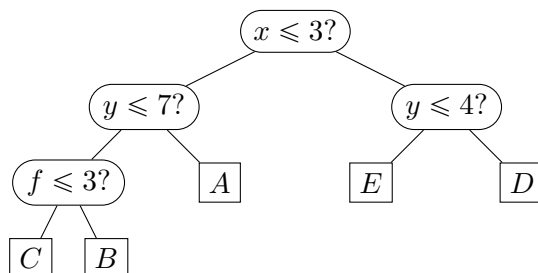
3	5	4	2	1	4	5	3
6	6	6	6	6	6	6	6
6	6	6	6	6	6	6	6
3	5	4	2	1	4	5	3

Kolikšna je višina (tj. število nivojev minus 1) KD-drevesa za takšno razporeditev?

- (d) Kako bi se spremenil odgovor pri prejšnji podnalogi, če bi točkam dodali še četrto dimenzijo (barvo  $b \in \{0, 1\}$ ) in bi ustrezno prilagodili gradnjo KD-drevesa, tako da bi po novem točke po koordinati  $x$ ,  $y$  oziroma  $f$  delili na nivojih s številkami, ki so enakovredne 0, 1 oziroma 2 po modulu 4, po barvi pa na nivojih s številkami, ki so enakovredne 3 po modulu 4?

## Rešitve

- (a)



- (b) Težave nastopijo, kadar skušamo dve ali več točk z isto vrednostjo določene koordinate deliti po tej isti koordinati. Problem lahko rešimo tako, da točko  $(x, y, f)$  pretvorimo v  $(\bar{x}, \bar{y}, \bar{f}) = (x|y|f, y|f|x, f|x|y)$ . Na ta način bo v primeru  $(x, y, f) < (x', y', f')$  veljalo, denimo,  $\bar{x} < \bar{x}'$  v vseh treh situacijah: (1)  $x < x'$ ;

(2)  $x = x' \wedge y < y'$ ; (3)  $x = x' \wedge y = y' \wedge f < f'$ . Podobno velja za primerjavo po koordinatah  $y$  in  $f$ .

- (c) Najgloblji listi so na nivoju  $\lceil \lg n \rceil$ . Seveda je tolikšna tudi višina drevesa. V našem primeru imamo  $\lceil \lg 32 \rceil = 5$ .
- (d) Odgovor se ne spremeni, saj je število nivojev drevesa odvisno zgolj od števila točk, ne pa od števila dimenzij, po katerih jih delimo. Vsaka delitev namreč razpolovi število obravnavanih točk, ne glede na to, po kateri dimenziji jih delimo.