

Domáca úloha č. 1

1-AIN-105, Zima 2022

Termín: 17.10.2022, 22:00

Skôr ako sa pustíte do riešenia domácej úlohy, oboznámte sa so všeobecnými pokynmi, ktoré sú priložené na konci tohto dokumentu. Riešenia, ktoré odovzdáte, musia byť vaše vlastné. Neopisujte a nesnažte sa nájsť riešenia v literatúre alebo na internete!

1. [20 bodov] Asymptotická zložitosť.

- a) Pre každý pár funkcií $f(n)$ a $g(n)$ v nasledujúcej tabuľke uveďte, v akom vzťahu sú tieto funkcie (vyplňte v každom políčku tabuľky áno alebo nie). V krátkosti zdôvodnite každú (aj zápornú) odpoveď.

$f(n)$	$g(n)$	$f(n) \in O(g(n))$	$f(n) \in \Theta(g(n))$	$f(n) \in \Omega(g(n))$
$n \log^{154} n$	$n^{1.1}$			
$n \sin \pi n $	\sqrt{n}			
$n^2 + n^{3/2} + n \log n$	$\binom{n}{2}$			
$8^{\log n}$	$n^3 \log n$			

- b) Pri písaní písomky ste narazili na podobný príklad ako v časti a), akurát funkcie $f(n)$ a $g(n)$ sú zložitosti algoritmov, ktoré si žiaľ nepamätáte. Za každú správnu odpoveď je 1 bod, za nesprávnu odpoveď je -1 bod. Aký je očakávaný počet získaných bodov, ak sa rozhodnete vypísať do tabuľky odpovede áno alebo nie náhodne?
- c) Ak v časti b) navyše predpokladáte, že $f(n) = n^k$ a $g(n) = n^\ell$ (pre nejaké konštanty k a ℓ), viete navrhnúť aj lepšiu stratégiu, ktorá bude mať vyšší očakávaný počet získaných bodov? Môžete predpokladať, že prípady $k < \ell$, $k > \ell$ alebo $k = \ell$ sú rovnako pravdepodobné.
2. [20 bodov] **Zábava s triedením.** Dané je pole $A[0..n-1]$, ktoré obsahuje n čísel (čísla sa neopakujú). Dané je tiež kladné číslo k . Uvažujme nasledujúci pseudokód:

```
sort array A by merge sort;
j=0; sum=0;
for(i=0; i<n; i++) {
  while(A[j]<A[i]-k) {
    j++;
  }
  while(A[j]==A[i]-k) {
    sum++; j++;
  }
}
```

- a) Stručne opíšte aký je výsledok pseudokódu, teda aká hodnota bude uložená v premennej `sum` po jeho skončení.
- b) Aká je časová zložitosť pseudokódu v najhoršom a priemernom prípade?
- c) Ako by sa zmenil výsledok vašej analýzy, ak by sme namiesto merge sortu použili quick sort?
3. [20 bodov] **Halda stupňa 7 (programátorská úloha).** Halda stupňa 7 vyzerá rovnako ako binárna, len každý vrchol má sedem synov. (Presnejšie, nanačvých sedem.) Halda má tvar takmer úplného stromu, pričom posledná úroveň stromu môže byť zaplnená len sčasti, a to zľava doprava. Aj túto haldu teda vieme uložiť po úrovniach do poľa, rovnako ako binárnu. Koreň bude uložený na pozícii 0. Synovia vrcholu uloženého na pozícii k budú uložení na pozíciách $7k+1$ až $7k+7$. Budeme uvažovať haldu s maximom navrchu, to znamená že každý vrchol musí mať aspoň takú veľkú hodnotu ako jeho synovia.

Samotné vkladanie prvkov do haldy aj výber z nej implementujeme tak ako na prednáške: Nový prvok pridáme ako nový list a následne použijeme funkciu SiftUp (uspôsobenú na novú štruktúru haldy).

Maximum vyberáme tak, že do koreňa presunieme hodnotu z posledného listu, posledný list odstránime a následne použijeme uspôsobenú funkciu Heapify. V tomto príklade budeme počítat počet výmen a počet porovnaní; všimnite si, že jedna iterácia funkcie SiftUp si vyžaduje jedno porovnanie, kým iterácia funkcie Heapify si vyžaduje toľko porovnaní koľko má aktuálny vrchol synov.

Vaše implementácie SiftUp a Heapify by mali zodpovedat nasledovnému pseudokódu:

SiftUp:

kým (nie som v koreni) a (aktuálny prvok je ostro väčší ako jeho otec):
vymeň aktuálny prvok s jeho otcom

Heapify:

kým (nie som v liste) a (aktuálny prvok je ostro menší ako niektorý z jeho synov):
vymeň aktuálny prvok s jeho najväčším synom
(v prípade rovnosti medzi synmi: najľavejším z najväčších synov)

Vstup. Na prvom riadku vstupu je číslo n - počet operácií vloženia/vybratia prvku. Nasleduje n niekoľko riadkov, kde každý riadok má buď formát vloz <cislo>, alebo vyber. Ak načítate riadok vloz <cislo>, tak máte do oboch hald vložiť číslo <cislo>. Po načítaní riadku vyber máte z haldy vybrať najväčší prvok. Je zaručené, že z haldy máte vždy čo vybrať (teda že nenastane situácia, že máte prázdnu haldu, a na vstupe bude operácia vyber).

Výstup. Pre každý beh vypíšte samostatný riadok so štyrmi číslami p_2 q_2 p_7 q_7 (čísla budú oddelené medzerami). Číslo p_2 je celkový počet porovnaní, ktoré sme spravili pri vkladaní prvkov do bežnej binárnej haldy. Číslo q_2 je celkový počet porovnaní, ktoré sme spravili pri vyberaní prvkov z binárnej haldy. Čísla p_7 a q_7 znamenajú to isté, ak použijeme haldu stupňa 7.

Príklady:

Vstup:

```
5
vloz 3
vyber
vloz 4
vyber
vloz -3
```

Výstup:

```
0 0 0 0
```

Vstup:

```
12
vloz 3
vloz 4
vloz 20
vloz 7
vloz 13
vloz 0
vyber
vyber
vyber
vyber
vyber
vyber
```

Výstup:

```
7 10 5 10
```

Poznámka: v druhom príklade sa postupne diali tieto porovnania:

```
operacia vloz
halda je: [3]
operacia vloz
halda je: [3]
porovnaniam: 0 0 0 0
```

```
operacia vloz
halda je: [3, 4]
porovnavam v 2-arnej indexy 0 a 1
operacia vloz
halda je: [3, 4]
porovnavam v 7-arnej indexy 0 a 1
porovnaniam: 1 0 1 0
```

```
operacia vloz
halda je: [4, 3, 20]
porovnavam v 2-arnej indexy 0 a 2
operacia vloz
halda je: [4, 3, 20]
porovnavam v 7-arnej indexy 0 a 2
porovnaniam: 2 0 2 0
```

```
operacia vloz
halda je: [20, 3, 4, 7]
porovnavam v 2-arnej indexy 1 a 3
porovnavam v 2-arnej indexy 0 a 1
operacia vloz
halda je: [20, 3, 4, 7]
porovnavam v 7-arnej indexy 0 a 3
porovnaniam: 4 0 3 0
```

```
operacia vloz
halda je: [20, 7, 4, 3, 13]
porovnavam v 2-arnej indexy 1 a 4
```

porovnavam v 2-arnej indexy 0 a 1
operacia vloz
halda je: [20, 3, 4, 7, 13]
porovnavam v 7-arnej indexy 0 a 4
porovnanania: 6 0 4 0

operacia vloz
halda je: [20, 13, 4, 3, 7, 0]
porovnavam v 2-arnej indexy 2 a 5
operacia vloz
halda je: [20, 3, 4, 7, 13, 0]
porovnavam v 7-arnej indexy 0 a 5
porovnanania: 7 0 5 0

operacia vyber
halda je: [0, 13, 4, 3, 7]
porovnavam v 2-arnej indexy 1 a 0
porovnavam v 2-arnej indexy 2 a 1
porovnavam v 2-arnej indexy 3 a 1
porovnavam v 2-arnej indexy 4 a 3
operacia vyber
halda je: [0, 3, 4, 7, 13]
porovnavam v 7-arnej indexy 1 a 0
porovnavam v 7-arnej indexy 2 a 1
porovnavam v 7-arnej indexy 3 a 2
porovnavam v 7-arnej indexy 4 a 3
porovnanania: 7 4 5 4

operacia vyber
halda je: [0, 7, 4, 3]
porovnavam v 2-arnej indexy 1 a 0
porovnavam v 2-arnej indexy 2 a 1
porovnavam v 2-arnej indexy 3 a 1
operacia vyber
halda je: [0, 3, 4, 7]
porovnavam v 7-arnej indexy 1 a 0
porovnavam v 7-arnej indexy 2 a 1
porovnavam v 7-arnej indexy 3 a 2
porovnanania: 7 7 5 7

operacia vyber
halda je: [0, 3, 4]
porovnavam v 2-arnej indexy 1 a 0
porovnavam v 2-arnej indexy 2 a 1
operacia vyber
halda je: [0, 3, 4]
porovnavam v 7-arnej indexy 1 a 0
porovnavam v 7-arnej indexy 2 a 1
porovnanania: 7 9 5 9

operacia vyber
halda je: [0, 3]
porovnavam v 2-arnej indexy 1 a 0
operacia vyber
halda je: [0, 3]
porovnavam v 7-arnej indexy 1 a 0

porovnaní: 7 10 5 10

operacia vyber

halda je: [0]

operacia vyber

halda je: [0]

porovnaní: 7 10 5 10

operacia vyber

halda je: []

operacia vyber

halda je: []

porovnaní: 7 10 5 10

Vstup:

10

vloz 8

vyber

vloz 0

vyber

vloz -3

vyber

vloz -2

vyber

vloz -7

vloz -4

Výstup:

1 0 1 0

Vstup:

11

vloz 1

vloz 2

vloz -8

vloz 7

vloz 3

vyber

vyber

vloz 1

vloz -2

vloz -1

vloz -6

Výstup:

11 5 8 5

Vstup:

34
vloz -10
vyber
vloz 4
vyber
vloz -28
vyber
vloz 25
vyber
vloz -8
vloz -28
vyber
vloz 4
vloz -22
vloz 20
vyber
vloz 33
vloz -17
vloz -6
vloz 33
vloz 22
vyber
vloz 10
vyber
vloz -19
vloz -3
vloz 18
vloz -28
vloz 27
vloz 20
vyber
vloz 27
vloz 7
vloz 4
vloz -34

Výstup:

35 15 24 25

Odvzdávanie: Odkaz na odovzdanie programátorskej úlohy : <https://testovac.ksp.sk/tasks/eads2022-du1a/>

4. [20 bodov] **Opravovanie midtermov (programátorská úloha).** Prišiel čas midtermov a študenti napísali písomku z EADŠ. Docent Vinař sa pustil do jej opravovania. Písomky opravoval postupne, ale trvalo mu to dosť dlho. Študenti začali byť netrpezliví a chceli vedieť, ako sa im darilo. Docent sa rozhodol, že nebude zverejňovať výsledky len časti študentov, lebo by to bolo nefér voči tým, ktorí ešte neboli opravení. Aby ale študentom vyhovel, rozhodol sa po každej opravenej písomke zverejniť medián doteraz získaných bodových hodnotení. Napíšte program, ktorý mu pomôže s hľadáním mediánu.

Vstup. Na vstupe sú dva riadky. Na prvom je číslo $1 \leq N \leq 10^5$, a na druhom postupnosť N čísel d_0, \dots, d_N ($0 \leq d_i < 10^9$). Číslo d_i je počet bodov i -teho opraveného midtermu.

Výstup. Na výstupe je N čísel oddelených medzerami, pričom i -te číslo je mediánom čísel d_0, \dots, d_i . Medián i čísel je $\lfloor i/2 \rfloor$ -hé najmenšie číslo. (Všimnite si, že číslujeme od nuly.) Za posledným číslom vypíšte znak konca nového riadku, a nie medzeru.

Príklady:

Vstup:

2
1000 100

Výstup:

1000 100

Vstup:

10
158 182 112 92 52 201 193 167 177 167

Výstup:

158 158 158 112 112 112 158 158 167 167

Vstup:

10
83 36 16 58 65 80 52 83 73 71

Výstup:

83 36 36 36 58 58 58 58 65 65

Odobzdvávanie: Odkaz na odovzdanie programátorskej úlohy : <https://testovac.ksp.sk/tasks/eads2022-du1b/>

Všeobecné pokyny

Písomné úlohy. Písomné úlohy odovzdávajte do *Google Classroom* ako PDF súbory v stanovenom termíne. **Každý príklad odovzdajte v osobitnom PDF súbore.** Na neskoro odovzdané riešenia sa nebude prihliadať. Píšte riešenia takým spôsobom, aby obsahovali všetku potrebnú informáciu na pochopenie vášho riešenia, ale súčasne aby boli stručné a ľahko pochopiteľné. Všetky tvrdenia je potrebné zdôvodniť (a to aj v prípade, že to nie je explicitne napísané v zadaní).

Ak sa v zadaní požaduje vyriešenie algoritmickej úlohy, odovzdajte najlepší algoritmus, aký viete navrhnúť. Základným kritériom na hodnotenie bude *správnosť algoritmu*, druhým kritériom bude jeho *časová, prípadne pamäťová zložitosť*. Správny ale pomalý algoritmus dostane podstatne viac bodov ako algoritmus, ktorý je síce rýchly, ale nedá správnu odpoveď na každý vstup. Neefektívne algoritmy spĺňajúce podmienky zadania dostanú cca 50% bodov. Súčasťou vášho riešenia musia byť nasledujúce časti:

- Najprv popíšte hlavnú myšlienku algoritmu.
- Vyjadrite algoritmus formou pseudokódu.
- Ak to nie je zrejmé na prvý pohľad, ukážte že váš algoritmus je správny.
- Nezapudnite na analýzu zložitosti algoritmu.

Ak nie je povedané inak, logaritmy majú základ 2.

Programátorské úlohy. Pri programátorských úlohách je Vašou úlohou odovzdať len funkčný program, nie je vyžadované písomné riešenie. Riešenie odovzdávate cez webové rozhranie <https://testovac.ksp.sk/tasks/>, kde bude okamžite otestované na niekoľkých vstupoch a dozviete sa, koľko bodov získalo (body získate, keď všetky vstupy z danej sady vyriešite správne v časovom limite). Riešenie môžete odovzdávať aj viackrát, hodnotí sa posledné riešenie odovzdané v stanovenom termíne. Na odovzdávanie riešení (keďže na testovač nefungujú univerzitné prihlasovacie údaje) je nutné sa na stránke zaregistrovať (vľavo na stránke testovača). Pri vytváraní účtu nastavte správne meno a priezvisko, a ako používateľské meno nastavte Váš univerzitný login. Nezapudnite tiež napísať Vaše používateľské meno do PDF súboru k ostatným úlohám, ktoré odovzdávate. Podrobnosti o tom, ako má váš program vyzeráť (vrátane povolených programovacích jazykov), nájdete v sekcii "Čo odovzdávať?". Informácie o testovači nájdete v sekcii "Odpovede testovača".

Na zoznámenie sa s rozhraním testovača si môžete vyskúšať naprogramovať niektoré z úloh z časti "Úvod do programovania".