



# NAPOJ

UČNA PRIPRAVA

Urejanje

DATUM

NAPOJ

## OSNOVNI PODATKI

Šola:
Letnik: 1. Letnik
Datum:
Predmet: Informatika
Učna tema: Urejanje podatkov
Učna enota: urejanje, koncepti urejanja
Učne oblike: <ul style="list-style-type: none"><li>- frontalno, individualno, skupinsko</li></ul>
Učne metode: <ul style="list-style-type: none"><li>- razlaga, pogovor, demonstracija, reševanje problemov</li></ul>
Predznanje: <ul style="list-style-type: none"><li>- poznajo pojme algoritem, funkcija, tabela, spremenljivka</li></ul>
Operativni učni cilji Ob koncu učne ure učenec: <ul style="list-style-type: none"><li>- Spozna problem urejanja podatkov</li><li>- Razume logično oziroma strukturno razmišljanje v proces urejanja</li><li>- Spozna različne koncepte urejanja</li><li>- Razume dobre in slabe lastnosti teh konceptov</li></ul>
Učna sredstva: <ul style="list-style-type: none"><li>- Učila: delovni list, računalnik</li><li>- Učni pripomočki: računalnik, Python, zvezek, tabla, tehtnica z različno težkimi utežmi oziroma podobno orodje</li></ul>
Didaktične etape učnega procesa: <ol style="list-style-type: none"><li>1. Pripravljanje ali uvajanje</li><li>2. Obravnava nove učne snovi ali usvajanje</li><li>3. Urjenje</li><li>4. Ponavljanje</li><li>5. Preverjanje</li></ol>
Medpredmetne povezave: slovenščina, angleščina
Literatura: <ul style="list-style-type: none"><li>- M. A. Weiss, Data Structures and Algorithm Analysis (2nd Edition), Addison Wesley, 1994.</li><li>- M. T. Goodrich, R. Tamassia, Data structures and algorithms in Java, 2001.</li><li>- G. Anželj, J. Brank, A. Brodnik, P. Bulić, M. Ciglarič, M. Đukić, L. Furst, M. Kikelj, A. Krapež, H. Medvešek, N. Mori, M. Pančur, P. Sterle, Računalništvo in informatika 1, Založba Univerze na Primorskem and Založba Fakultete za računalništvo in informatiko and Založba Fakultete za elektrotehniko and računalništvo in informatiko, 2015. URL <a href="https://lusy.fri.uni-lj.si/ucbenik/book/index.html">https://lusy.fri.uni-lj.si/ucbenik/book/index.html</a>.</li><li>- B. Vocking, H. Alt, M. Dietzfelbinger, R. Reischuk, C. Scheideler, H. Vollmer, D. Wagner, Algorithms unplugged, Springer Science &amp; Business Media, 2010.</li></ul>

<ul style="list-style-type: none"> <li>- Computer Science Unplugged, Lightest and Heaviest – Sorting Algorithms. Dostopno na: <a href="https://classic.csunplugged.org/wp-content/uploads/2014/12/unplugged-07-sorting_algorithms.pdf">https://classic.csunplugged.org/wp-content/uploads/2014/12/unplugged-07-sorting_algorithms.pdf</a> (pridobljeno 2. 8. 2018).</li> <li>- E. Knuth D, The art of computer programming, fundamental algorithms (1980).</li> <li>- D. E. Knuth, The art of computer programming: sorting and searching, Vol. 3, Pearson Education, 1997.</li> <li>- Tutorialspoint, Data structures - merge sort algorithm. Dostopno na: <a href="https://www.tutorialspoint.com/data_structures_algorithms/merge_sort_algorithm.htm">https://www.tutorialspoint.com/data_structures_algorithms/merge_sort_algorithm.htm</a> (pridobljeno 2. 8. 2018)</li> <li>- Leder, Misha, Presentation 4: sorting. Dostopno na: <a href="https://sites.google.com/site/childrenandtechnology/Home/presentation-4-sorting">https://sites.google.com/site/childrenandtechnology/Home/presentation-4-sorting</a> (pridobljeno 2. 8. 2018).</li> <li>- J. Demšar, Urejanje, Dostopno na: <a href="http://vidra.si/urejanje">http://vidra.si/urejanje</a>, (pridobljeno 3. 8. 2018).</li> </ul>
Novi pojmi: <ul style="list-style-type: none"> <li>- urejanje z izbiranjem, urejanje z zlivanjem, hitro urejanje</li> </ul>
Priloga: <ul style="list-style-type: none"> <li>- Tabela z rezultati</li> </ul>

## POTEK UČNE URE

### UVODNI DEL: UVAJANJE

ČAS	UČITELJ	UČENEC	UČNE OBLIKE, METODE, TEHNIKE, UČNI PRIPOMOČKI
<b>Uvod</b> 5 min	<i>Učitelj pozdravi učence, zapiše manjkajoče učence in po potrebi prosi dežurnega učenca, da pobriše tablo.</i>	Odzdravijo, naštejejo manjkajoče učence, dežurni učenec pobriše tablo.	Frontalno.
<b>Motivacija</b> 10 min	Danes se bomo posvetili urejanju. Urejanje je problem, pri katerem hočemo elemente nekega zaporedja prerazporediti v željen vrstni red. Večina ljudi imajo radi stvari urejene oziroma pospravljene. Mi lahko naštejete nekaj takih primerov in razložite, zakaj je v teh primerih dobro imeti stvari urejene? Na primer: <ul style="list-style-type: none"> <li>• Osebe v imeniku so urejene po abecedi, da hitreje najdemo želeno.</li> <li>• Pri prenosu olimpijskih iger se na koncu teka izpišejo imena tekmovalcev, ki so urejeni po času, da gledalci lažje vidimo kdo je zmagal.</li> <li>• Knjige na polici so urejene ali po naslovu,</li> </ul>	Poslušajo.  Odgovorijo, naštejejo primere.	Frontalno, razlaga, pogovor.

	<p>ali po pisatelju, spet z namenom da hitreje najdemo želeno.</p> <p>Naslednje vprašanje za vas je, po katerih kriterijih lahko urejamo elemente? Ali je primerjanje števil edini možen kriterij? Drugi možni kriteriji so na primer še:</p> <ul style="list-style-type: none"> <li>• Odtенок barve</li> <li>• Oblika predmetov</li> <li>• Veselje (nekateri čustvenčki so bolj, nekateri manj veseli)</li> <li>• Moč zvoka (ali je električna kitara glasnejša od orglic)</li> <li>• Mehkosti predmeta</li> </ul> <p>Pojdimo raje iz druge smeri: naštejte nekaj stvari, ki jih ne moremo urejati. <i>Učitelj odgovarja na primere, na kakšen način se jih da urediti.</i> Vidimo torej, da je veliko več kriterij za urejanje kot pa le po velikosti števil ter da skoraj ni stvari, katere se ne bi dalo urediti glede na nek kriterij.</p>	<p>Odgovorijo, naštejejo druge kriterije.</p> <p>Poskušajo naštetiti stvari, ki se jih ne da urediti po nekem kriteriju.</p>	
--	---	--	--

## GLAVNI DEL: OBRAVNAVANJE UČNE SNOVI

VSEBINSKI POUDARKI	UČITELJ	UČENEC	UČNE OBLIKE, METODE, TEHNIKE, UČNI PRIPOMOČKI
<p><b>Uvod v urejanje</b></p> <p>15 min</p>	<p>Danes bomo spoznali 3 koncepte urejanja. Še prej pa sem pripravil pripomočke za današnjo uro. Z njimi se bomo tudi spoznali s temi koncepti. Vidimo, da imam tehtnice in lončke, ki so različno težki.</p> <p><i>Opomba učitelju: izbira ustreznih učnih pripomočkov (v tej učni pripravi so uporabljeni tehtnica in različno težki lončki (lahko bi uporabili namesto tehtnice kar roke)) je prepuščena učitelju. Važno je, da učenci ne morejo urediti elementov z metodo ostrega pogleda, saj se s tem izgubi razumevanje, kako računalnik primerja elemente.</i></p> <p>Vaša naloga je, da se razporedite v skupine. Vsaka skupina bo dobila eno tehtnico in nekaj različno težkih lončkov. Eden iz skupine jih bo moral urediti glede na težo od najlažjega do najtežjega, drug učenec iz skupine bo pa zapisoval število primerjanj. Naj vas opozorim, da lahko hkrati primerjate zgolj 2 lončka, saj boste tako boljše kako</p>	<p>Spremljajo razlago, si zapisujejo.</p> <p>Učenci se razdelijo v skupine in uredijo lončke po teži.</p>	<p>Frontalno, razlaga, skupinsko delo.</p>

	<p>deluje računalnik. Pa kar začnite.</p> <p><i>Učitelj medtem na tablo nariše tabelo iz priloge 1.</i></p> <p>Prosim, da mi sedaj vsaka skupina pove, koliko primerjanj je potrebovala za ureditev lončkov.</p> <p><i>Učitelj zapiše številke v tabelo pod »prosto urejanje«.</i></p> <p>Čestitke najboljši ekipi! Nam lahko zaupate, po kakšnem sistemu ste karte uredili? Ste imeli težave z izbiranjem sistema ali ste jih uredili intuitivno? Kaj pa ostale skupine, ste uredili karte na isti način?</p> <p>Kot vidimo je izbira dobrega načina urejanja oziroma algoritma pomembna. Za urejanje lahko izbiramo med različnimi algoritmi, pri čemer se ti razlikujejo po preprostosti implementacije in hitrosti urejanja. Danes bomo spoznali enega enostavnejša, a počasnejšega algoritma – urejanje z izbiranjem, ter 2 hitrejša, a bolj zapletena – urejanje z zlivanjem in hitro urejanje.</p> <p>Uro bomo nadaljevali tako, da bom za vsak algoritem razložil njegovo delovanje, vsaka skupina pa bo z uporabo teh algoritmov uredila lončke po teži, pri čemer boste seveda prešteli število primerjav. Te bom nato napisal na tablo, da bomo na koncu videli, kateri algoritmi so boljši in kateri slabši. Boljši so seveda tisti z manjšim številom primerjanj. Vsak algoritem bomo tudi implementirali v Pythonu ter poiskali časovno zahtevnost.</p>	<p>Skupine sporočijo število primerjanj.</p> <p>Učenci odgovorijo glede na lastno izkušnjo.</p> <p>Učenci poslušajo in si zapisujejo.</p>	
<p><b>Urejanje z izbiranjem</b></p> <p>10 minut</p>	<p>Začnimo z urejanjem z izbiranjem. Ta algoritem je zelo enostaven: vsak obhod elementov poiščemo najmanjšega oziroma najlažjega in ga postavimo na začetek vrste. Kako? Vzamemo prvi lonček in ga primerjamo z naslednjim. Če je naslednji lažji, je sedaj ta najmanjši in njega primerjamo z nadaljnjimi lončki. Ko pridemo do konca, ta lonček postavimo na začetek vrste in ponovimo postopek z naslednjim lončkom v vrsti.</p> <p>Vzemimo na primer zaporedje [5,8,7,2,3,10,9]. Postopek urejanja je sledeč:</p> <p>[2,5,8,7,3,10,9] Se sprehodimo čez elemente, najdemo najmanjšega, tj. 2, ter ga postavimo na začetek.</p> <p>[2,3,5,8,7,10,9] najmanjše je 3, jo postavimo na indeks 1</p> <p>[2,3,5,7,8,10,9] najmanjše je 5, jo postavimo na indeks 2</p>	<p>Spremljajo razlago in si zapišejo v zvezek.</p>	<p>Frontalno, razlaga, pogovor, skupinsko, demonstracija.</p>

	<p>[2,3,5,7,8,9,10] najmanjše je 7, jo postavimo na indeks 3</p> <p>[2,3,5,7,8,9,10] najmanjše je 8, jo postavimo na indeks 4</p> <p>[2,3,5,7,8,9,10] najmanjše je 9, jo postavimo na indeks 5</p> <p>[2,3,5,7,8,9,10] najmanjše je 10, jo postavimo na indeks 6</p> <p>Sedaj je na vas, da uredite lončke, in ne pozabite prešteti število primerjanj.</p> <p>Sedaj vas prosim, da mi sporočite, koliko primerjanj ste dobili. <i>Učitelj izpolni v tabeli stolpec Urejanje z izbiranjem.</i> Kaj mislite, zakaj ste vsi dobili enako število primerjav? Tako, ker ob vsakem obhodu vedno obiščemo vse karte. Zapišimo algoritem v Pythonu. <i>Učitelj razloži kodo.</i></p> <pre>def uredi(sez):     for i in range(len(sez)):         iMin = i         for j in range(i+1, len(sez)):             if sez[j] &lt; sez[iMin]:                 iMin = j         sez[i], sez[iMin] = sez[iMin], sez[i]     return sez</pre>	<p>Skupine uredijo lončke z uporabo urejanja z izbiranjem.</p> <p>Skupine sporočijo število primerjanj in odgovorijo na vprašanja.</p>	
<p><b>Urejanje z zlivanjem</b></p> <p>18 min</p>	<p>Pojdimo na urejanje z zlivanjem. Tu pridemo do zanimivega koncepta, ki se imenuje deli in vladaj. Pri tem razdelimo problem na dva podproblema, ju rešimo, ter ju združimo v rešitev prvotnega problema. Urejanje z zlivanjem poteka takole:</p> <ul style="list-style-type: none"> <li>• 1. del (deli): razdeli lončke na dva enaka dela, vsako del tudi na pol, vsako četrtino tudi na pol, ... dokler ne dobimo, da je vsak del velik po 0 ali 1 element.</li> <li>• 2. del (vladaj): združujemo po dva dela tako, da so v novem seznamu lončki urejeni.</li> </ul> <p>Za lažje razumevanje si na primeru. Imamo seznam [7,4,5,2]. Najprej ga razdelimo na pol, vsak del pa še na pol. Tako dobimo sezname [7], [4], [5] in [2]. Nato združimo prva dva seznama tako, da primerjamo prvi element iz obeh seznamov ter manjšega prenesemo v združen seznam. To delamo, dokler imamo še kaj elementov v prvem</p>	<p>Sledijo razlagi in si zapisujejo.</p>	<p>Frontalno, razlaga, pogovor, skupinsko, demonstracija.</p>

ali drugem seznamu.

Sez1	Sez2	združenSez
[7]	[4]	[]
[7]	[]	[4]
[]	[]	[4,7]

Enako storimo z naslednjima dvema podseznama.

Sez1	Sez2	združenSez
[5]	[2]	[]
[5]	[]	[2]
[]	[]	[5,2]

Na koncu združimo še ta dva seznama.

Sez1	Sez2	združenSez
[4,7]	[5,2]	[]
[4,7]	[5]	[2]
[7]	[5]	[2,4]
[7]	[]	[2,4,5]
[]	[]	[2,4,5,7]

Uredite sedaj lončke z uporabo urejanja z zlivanjem, pa ne pozabite prešteti število primerjanj.

Sedaj vas prosim, da mi poveste koliko primerjanj ste dobili. Kaj mislite, zakaj ste dobili enako število primerjav?

Sedaj zapišimo algoritem v Pythonu.

```
def zlijZaporedji(sez1, sez2):
    zlito = []
    i = 0
    j = 0

    while i < len(sez1) or j <
len(sez2):
        if j >= len(sez2):
            zlito.append(sez1[i])
```

Skupine uredijo lončke z uporabo urejanja z zlivanjem.

Skupine sporočijo število primerjanj in odgovorijo na vprašanja.

	<pre> i += 1 elif i &gt;= len(sez1):     zlito.append(sez2[j])     j += 1 elif sez1[i] &lt; sez2[j]:     zlito.append(sez1[i])     i += 1 else:     zlito.append(sez2[j])     j += 1 return zlito  def UrejanjeZZlivanjem(sez):     n = len(sez)     if n &lt;= 1:         return sez      levo = sez[:n // 2]     desno = sez[n // 2:]      levoUrejeno = UrejanjeZZlivanjem(levo)     desnoUrejeno = UrejanjeZZlivanjem(desno)     združeno = zlijZaporedji(levoUrejeno, desnoUrejeno)     return združeno </pre> <p><i>Učitelj razloži kodo.</i></p> <p>Funkcija <code>UrejanjeZZlivanjem()</code> razdeli prejet seznam na podseznama ter kliče funkcijo <code>zlijZaporedji()</code> in mu poda 2 taka podseznama, funkcija pa ju združi.</p>		
<p><b>Hitro urejanje</b></p> <p>17 min</p>	<p>Pojdimo še na hitro urejanje oziroma po angleško Quicksort. Tudi ta algoritem deluje po načinu deli in vladaj, a na malo drugačen način. Na začetku vzamemo prvi element, ki ga imenujemo pivot. Nato gremo čez vse ostale elemente in jih razporedimo v nove sezname glede na to, ali so manjši ali večji od pivota. Če je element manjši, gre v prvi seznam (levo od pivota), če večji, pa v drugega (desno od pivota). Postopke ponovimo z obema kupoma. Primer:</p> <p>[4,2,5,7,3,1,8] vzamemo 4 za pivot, v levi kup postavimo manjše, v desni pa večje elemente</p> <p>[2,3,1], [4], [5,7,8] iz levega kupa vzamemo 2 za pivot, iz desnega pa 5</p> <p>[1], [2], [3], [4], [], [5], [7,8] iz skrajno desnega vzamemo za pivot 7</p> <p>[1], [2], [3], [4], [], [5], [7], [8] združimo sezname</p> <p>[1,2,3,4,5,7,8] rešitev</p>	<p>Sledijo razlagi in si zapisujejo.</p>	<p>Frontalno, razlaga, pogovor, skupinsko, demonstracija.</p>



	<p>Pa kar začnete z urejanjem, in ne pozabite prešteti število primerjanj.</p> <p>Sedaj vas prosim, da mi poveste koliko primerjanj ste dobili. Zapišimo algoritem v obliki psevdokode:</p> <pre>def hitro_uredi(sez):     manjsi = []     enaki = []     vecji = []      if len(sez) &gt; 1:         pivot = sez[0]         for st in sez:             if st &lt; pivot:                 manjsi.append(st)             elif st == pivot:                 enaki.append(st)             else:                 vecji.append(st)         leviSez = hitro_uredi(manjsi)         desniSez = hitro_uredi(vecji)         return leviSez + enaki + desniSez     else:         return sez</pre> <p><i>Učitelj razloži kodo. Funkcija hitro_uredi() razdeli elemente prejetega seznama v 3 sezname ter s pomočjo rekurzije uredi levi in desni seznam.</i></p>	<p>Skupine uredijo lončke z uporab urejanja z zlivanjem.</p> <p>Skupine sporočijo število primerjanj in odgovorijo na vprašanja.</p>	
<p><b>Časovna zahtevnost</b></p> <p>10 min</p>	<p>Kot ste opazili, je urejanje z izbiranjem enostavnejše kot urejanje z zlivanje ali hitro urejanje. Kaj pa glede časovne zahtevnosti? Kateri algoritem je najhitrejši? Mogoče nam je že iz števila primerjanj jasno, a to ne nujno kaže realne slike, zato uporabimo naše znanje iz poglavja Poraba časa in časovna zahtevnost.</p> <p>Če pogledamo kodo urejanja z izbiranjem, vidimo, da imamo for zanko, znotraj nje pa še eno for zanko, pri čemer gresta obe preko vseh elementov seznama. Kakšna je torej časovna zahtevnost algoritma?</p> <p>Pri urejanju z zlivanjem moramo malo razmisliti. Algoritem se vsakič rekurzivno kliče, a pri tem vedno razdeli seznam na dve polovici. Tako imamo ob prvem klicu 2 polovici, ob drugem klicu 4 četrtine, ob tretjem klicu 8 osmin, itd., dokler ne pridemo do seznamov velikosti 1 ali 0. Temu pravimo dvojiški logaritem. Za delitev torej potrebujemo <math>\log(n)</math> časa. Kaj pa za združevanje seznamov? Ker pri tem pregledamo vse elemente iz obe seznamov, je časovna zahtevnost kar <math>n</math>. Skupaj je torej časovna zahtevnost urejanja z</p>	<p>Sledijo razlagi in si zapisujejo.</p> <p>Odgovorijo: <math>O(n^2)</math>.</p>	<p>Frontalno, razlaga, pogovor.</p>

	<p>zlivanjem <math>O(n \cdot \log(n))</math>.</p> <p>Hitro urejanje deluje na podoben način. Tudi tu razdelimo seznam glede na pivot na 2 podseznama ter ju z rekurzijo uredimo. Časovna zahtevnost je tudi tukaj <math>O(n \cdot \log(n))</math>.</p>		
--	--	--	--

## ZAKLJUČNI DEL: ZAKLJUČNO PONAVLJANJE / PREVERJANJE

ČAS	UČITELJ	UČENEC	UČNE OBLIKE, METODE, TEHNIKE, UČNI PRIPOMOČKI
5 min	<p>Tako, spoznali smo 3 algoritme za urejanje elementov, pri čemer je eden počasnejši a enostavnejši, druga dva pa hitrejša, a zapletenejša. Kot povsod v življenju ima torej vsaka stvar dobre in slabe strani.</p> <p><i>Učitelj se zahvali učencem za sodelovanje, jim da napotke za domačo nalogo. Dežurni učenec naj po potrebi pobriše tablo. Učenci naj ugasnejo računalnike.</i></p>	<p>Poslušajo.</p> <p>Poslušajo, ugasnejo računalnike, dežurni učenec pobriše tablo.</p>	Frontalno.

## Priloge

### Priloga 1: Tabela z rezultati

	Prosto urejanje	Urejanje z izbiranjem	Urejanje z zlivanjem	Hitro urejanje
[Ime_skupine 1]				
[Ime_skupine 2]				
...				
[Ime_skupine n]				