

Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko



Ljubljana,
5. 4. 2017

Katalog znanj in vzorci
nalog za izbirni izpit za
vpis na magistrski študij
Pedagoško
računalništvo in
informatika
2017/2018



KATALOG ZNANJ ZA IZBIRNI IZPIT ZA VPIS NA MAGISTRSKI ŠTUDIJ PEDAGOŠKO RAČUNALNIŠTVO IN INFORMATIKA

Katalog znanj povzema znanja, ki se zahtevajo ob vpisu na magistrski študij Pedagoško računalništvo in informatika.

Dokument navaja področja, ki so zajeta v vprašanjih na izbirnem izpitu, ki bo v primeru omejitve vpisa osnova za odločitev o izbranih kandidatih. V primeru omejitve vpisa, bodo kandidati izbrani glede na:

- povprečno oceno doseženo na študiju I. stopnje (60 % točk),
- rezultate izbirnega izpita (40 %).

Programiranje

- osnove objektno-usmerjenega in proceduralnega programiranja
- programski konstrukti za kontrolo toka (vejanja, zanke, programske strukture)
- iteracija in rekurzija
- osnovni podatkovni tipi (cela števila, realna števila, nizi, logične vrednosti) in operatorji
- obravnava izjem (angl. exception handling)

Magnus L. Hetland: Beginning Python, 2nd ed., Apress, 2008.

Algoritmi

- osnove računske zahtevnosti algoritmov (notacija veliki O)
- podatkovne strukture (seznam, vrsta, sklad, množica, prioriteta vrsta, razpršena tabela, drevesa, grafi)
- nekaj metod razvoja programov (požrešno preiskovanje, deli in vladaj, dinamično programiranje)
- klasični algoritmi (urejanje, iskanje minimalnih vpetih dreves, iskanje najkrajših poti itd.)

Kononenko in sod. Programiranje in algoritmi, Založba FE & FRI, 2008. ali Cormen et al.: Introduction to algorithms, 3rd ed., The MIT press, 2009, poglavja 2, 3.1, 4.1, 7.1, 7.2, 10.1, 10.2, 11.2, 12.1, 12.2, 12.3, 15.1, 16.1, 22.1, 22.2, 22.3, 22.4., 23.1, 23. 2.

Računalniški sistemi

- osnove digitalnih vezij (Boolova algebra, kombinacijska in sekvenčna logika, končni avtomati)
- predstavitev števil v računalniku
- osnove računalniške arhitekture

Kodek, Dušan: Arhitektura in organizacija računalniških sistemov, BI-TIM, Ljubljana, 2008 (ISBN 978961-6046-08-4)



Didaktika in razvojna psihologija

- dejavniki in zakonitosti duševnega razvoja
- psihološke značilnosti šolarja, adolescenta in odraslega
- individualne razlike in njegov vpliv na razvoj posameznika in na njegovo uspešnost v šoli

Jacobson, D., Eggen, P., Kauchak, D. (1993). *Methods for Teaching. A Skills Approach*. Maxwell Macmillan International, New York, Oxford, Singapore, Sydney, str. 25-261 in 307-351. Marentič Požarnik, B. (2000). *Psihologija učenja in pouka*. Ljubljana: DZS. (280 str.)

Načrtovanje, izvajanje, spremljanje in vrednotenje vzgojno-izobraževalnega procesa

- taksonomije učnih ciljev
- didaktične komponente učnega procesa: od uvajanja do preverjanja in ocenjevanja
- učne metode. Učne oblike (individualno delo, skupinsko-kooperativno delo, tandem, frontalno delo)
- diferenciacija in individualizacija pouka
- nekatere strategije pouka: raziskovalni pouk, problemski pouk, projektno učno delo

Jacobson, D., Eggen, P., Kauchak, D. (1993). *Methods for Teaching. A Skills Approach*. Maxwell Macmillan International, New York, Oxford, Singapore, Sydney, str. 25-261 in 307-351. Marentič Požarnik, B. (2000). *Psihologija učenja in pouka*. Ljubljana: DZS. (280 str.)

V Ljubljani, 5. 4. 2017

Vzorci nalog za izbirni izpit za vpis na magistrski študij Pedagoško računalništvo in informatika

1. Dana je funkcija

```
int f (int a , int b) {  
    if (a > 0)  
        return f (a - 1 , b - 3);  
    else return b ;  
}
```

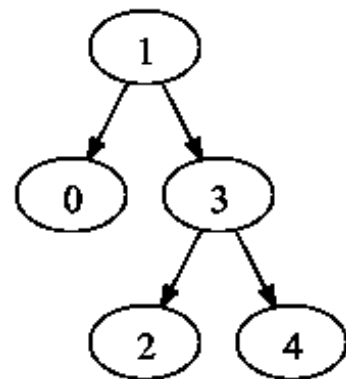
Kakšna je zgornja meja časovne zahtevnosti te funkcije?

- (a) $O(a)$
- (b) $O(ab)$
- (c) $O(a^2)$
- (d) $O(b^3)$
- (e) $O(a^b)$
- (f) $O(a/b)$

2. Imamo podano podatkovno strukturo za binarno iskalno drevo in spodnji algoritem.

Kakšen odgovor vrne algoritem, če mu na vhod damo drevo s slike?

- (a) 1
- (b) 0
- (c) 10
- (d) 4
- (e) 8
- (f) algoritem se zacikla



// podatkovna struktura

```
class BinTreeNode {  
    int element;  
    BinTreeNode left , right;  
}
```

// algoritem

```
int what(BinTreeNode n) {  
    if (n == null )  
        return 0;  
    else return n. element + what(n.left) + what(n.right);  
}
```

3. Katera od naslednjih trditev ni pravilna?

- (a) vsak rekurziven program je možno prevesti v iterativnega
- (b) izvajanje rekurzivnih programov je običajno počasnejše od iterativnih
- (c) repno rekurzijo lahko preprosto nadomestimo z zanko
- (d) interpretirani programski jeziki ne morejo izvajati rekurzivnih programov

4. Podan je spodnji program, ki sešteje vsako drugo število v seznamu števil. Na dveh označenih mestih v programu manjkata dva dela programske kode, ki sta označena z XXX in YYY:

```
# sums up every second element in the
# list "list", starting with element "start"

def sum2(list, start):
    if XXX > len(list)-1:
        return YYY
    else:
        return list[start] + sum2(list, start+2)

print(sum2([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 0))
```

Katera od naslednjih dopolnitev mest XXX in YYY je pravilna, da program deluje, kot prikazano?

- (a) XXX = start+2, YYY = 0
- (b) XXX = start+2, YYY = list[start]
- (c) XXX = start, YYY = 0
- (d) XXX = start, YYY = list[start]

5. katero desetiško število predstavlja dvojiško število 00111100?

- (a) 30
- (b) 60
- (c) 90
- (d) 120

6. Določite $f(0,0,0)$, $f(0,1,1)$ in $f(1,1,1)$.

- (a) 0,0,0
- (b) 0,1,0
- (c) 1,1,0
- (d) 1,1,1

