

Algoritmi i strukture podataka

2. blic – pitanja s foruma

ak. god. 2005/06

1. Razlika između deep/shallow kopiranja objekta...

2. Kako izgleda niz nakon shellsort $h=3$, nakon prvog koraka

3. Koja je sljedeća zamjena koju će bubble sort napraviti na zadanom nizu (nakon prvih par zadanih zamjena)

4. Stog ostvarem statičkim poljem od n elemenata može: **primiti n elemenata**

5. Netočna tvrdnja: **Prazan stog je greška u programu**

6. Koliko puta se zovu konstruktor i destruktork objekta koji je stvoren jednom u `main()`-u naredbom *objekt ime*;

i onda opet u funkciji koja se poziva iz `main()`-a a koja sadržava naredbu:

*objekt *ime=new objekt;*

Konstruktor se zove dva puta, a destruktork jednom, jer se ne koristi naredba delete za drugi objekt (prvi se pobriše automatski kad se napusti main())

7. Stog punimo sa for petljom $i=1$ do 10, onda ga praznimo $i=1$ do 5, sto je ostalo?

Rjesenje: 1,2,3,4,5

8. Napisan je kod nekog sorta u C-u, koji je to sort? Meni je bio **insertion...**

9. Što je na stogu nakon `push(push(push(5)+2))`?

`push` vraća 1 ako je uspio, 0 ako nije, stog može primiti barem tri elementa

Rjesenje: 5 3 1

10. Imamo klasu student koja ima konstruktor koji prima integer. Taj integer se pohrani u člansku varijablu `_X` i nakon toga se ispise slovo "A" `_X` puta. U destruktorku se ispise slovo "B" isto `_X` puta. Što će ispisati program:

```
main() { ...
```

```
student novi(2);
```

```
student *novi2=new student(3);
```

```
... }
```

Rjesenje: Ispisat će 5 slova "A" i 2 slova "B" jer se ne poziva destruktork za ovaj drugi objekt (ne koristi se delete)

11. Imamo klasu koja nema destruktork, a u konstruktoru se stvara polje (članska varijabla) pomoću naredbe **new**. Ispravna tvrdnja o toj klasi je **da se prostor koji zauzima polje neće osloboditi nakon brisanja objekta.**

12. Najveće memorijske zahtjeve ima **merge sort**

13. Pronaći stvarni medijan u sljedećem nizu

4 6 3 9 11 15 17

Rjesenje: 9, samo se pronađe broj koji bi bio na polovici niza da je on sortiran

14. Ako klasa "K" ima metodu "*metoda*" koja nije statična i ako imamo naredbu
K novi;

kako pozivamo metodu "metoda"?

Rješenje: novi.metoda();

15. Kako se inicijalizira referenca?

Rješenje: int a; &ref=a;

16. Stog iz starih blitzeva gdje je **rješenje 3 1 2**

17. Klasa trokut u kojoj su definirana dva konstruktora (s parametrom i bez-ovaj ima a=1 b=1 c=1) i funkcija *opseg*. U main()-u se definira *trokut t* i poziva se funkcija *opseg=t.opseg()*; uglavnom, **to poziva default konstruktor i opseg je 3**

18. Definiran je neki copy konstruktor za kompleksne brojeve i pita se sto radi this->re...

treba kliknuti na ono gdje piše da to ovisi o tome iz otkud je pozvan i za koje parametre...

19. Složenost bubble sorta – n^2

20. Shell sort s razmakom 3 za niz 66 88 99 22 77 55 33 11.

Rješenje: 22 11 99 33 77 55 33 88

21. Insertion sort na nekom čudnom primjeru gdje se sve ponavlja po dva puta...

22. Definicija funkcije dodaj za stog...

23. Funkcije *dodaj* i *skini* **uvijek imaju istu složenost bez obzira na broj članova.**

24. Bio je jedan koji stavlja na stog od 0 do 10 kao, al' veličina stoga je 5 i onda printf-a ono što popa...

Rješenje: 4 3 2 1 0

25. Napisan neki, sa stogom i sad, MAXSTOG je 10, on stavlja elemente na stog i pitanje je koliko najviše može staviti? A caka je u tome što mu je vrh definiran kao 0 a ne -1 što znači da je gore već jedan element.

Može se staviti 9

26. Iz OOPa je bio jeda GADAN, hrpetina koda, klasa, 2 članske funkcije (nisu komplicirane ali hey..) i onda glavni program i još sa nekim pokazivačima na tu klasu ovo-ono.. uglavnom.. gadno.. pita se za neki **y, m i pokazivač** ne znam kak se zvao...

NEMOJTE ME DRŽAT ZA RIJEČ al mislim da je bio točan onaj di je y=2

(obj_m=1, m=0, y=2)

27. Onda napisana definicija shallow copy i ponuđeni 5 različitih copy, trebalo staviti točni

28. Stavi na stog deklaracija funkcije (iz starih blitzeva)

29. Jedan iz referenci, kod ide otprilike...

```
int i=1;
```

```
int &referenca=i;
```

```
printf("%d %d", i, referenca);
```

```
i=2;
```

```
printf("%d %d", i, referenca);
```

30. Koja je apriorna složenost shell sorta

Rješenje: ovisi o hk

31. Zadano 5 koraka sortiranja i pita koji je to sort.

32. Najmanje je uputno **korisiti BUBBLE SORT kod velikih polja.**

33. Imamo funkcije `int push(int elem)` i `int pull()` koje rade sa stogom. Na stogu postoje već neki elementi (nije prazan). Funkcija `push` vraća 1 ako je uspjela, 0 inače, a funkcija `pull` vraća element kojeg skine sa stoga i ne vraća ništa za grešku.

Što će biti na stogu nakon naredbe `push(push(push(5)) + pull());`

Rješenje : 5 2

34. Imamo destruktora `~Stog()`; u nekoj definiranoj klasi `Stog`. Ako u glavnom programu instanciramo objekt na način `Stog obj;` kako će se taj objekt uništiti ili tako nešto!

Rješenje: AUTOMATSKI će se destruktora pozvati kada se izađe iz glavne funkcije i na taj način će se uništiti objekt.

35. Rečeno je da se treba sortirati selection sortom polje 3 5 1 4 2 i onda se pita koji koraci odgovaraju tom sortu:

malo je zbunjeno napisano, ali točan je odgovor (*ovaj prvi redak su ponovo napisali*):

Rješenje: **3 5 1 4 2**
 1 5 3 4 2
 1 2 3 4 5
 1 2 3 4 5
 1 2 3 4 5

36. Zadan je kod:

```
int a;
```

```
int &referenca = a;
```

```
a=1;
```

```
printf("%d %d ", a, referenca);
```

```
referenca=2;
```

```
printf("%d %d", referenca, a);
```

Što će se ispisati?

Rješenje: 1 1 2 2

37. Kad vam daju niz brojeva 2,4,6,15,9,5,1, MEDIJAN (NE STOŽER!!!) je 5, a ne 2, razlika je u onoj funkciji sa predavanja gdje ako po njoj radite, NE RADITE ZADNJI KORAK, Zamijeni(`&polje[sredina]`, `&polje[desno-1]`), dakle, još jednom, to se ne radi, već se **samo medijan klikne, u ovom slučaju 5**