

## Osnove algoritmov in podatkovnih struktur 2, PISNI IZPIT, 12.6. 2000

Literatura je dovoljena. Naloge so enakovredne. Čas pisanja 75 min.

Komentirajte programe! Obvezno pisanje v jezikih pascal, modula ali oberon!

1. Dan je algoritem:

```
procedure p(n,m : integer);
var i,j : integer;
begin
  if n < m then begin
    for i := 1 to m do for j := 1 to n do writeln(i,' ',j);
    p(n+1, m - 2)
  end
end;
```

- (a) Oceni časovno zahtevnost tega algoritma.  
(b) Neobvezno (za dodatnih 10 točk): Spremeni algoritem p v iterativnega.
2. Kakšen rezultat vrnejo in kakšna je časovna zahtevnost izvajanja, če izvajaš naslednje algoritme na grafu, ki je pot (kakšne omejitve, če sploh kakšne, morajo veljati za pot?):
- (a) Kruskalov algoritem  
(b) Primov algoritem  
(c) algoritem Dijkstra  
(d) algoritem za iskanje Kritične poti
3. Dani so elementi, katerih ključi so besede, in njihova verjetnostna porazdelitev. Definiraj urejenost in zgradi optimalno binarno iskalno drevo ob predpostavki, da so verjetnosti iskanja vseh vmesnih elementov enake 0.05, razen za elemente, ki so večji od največjega elementa v drevesu, katerih verjetnost iskanja je 0.1:

element	velik	srednji	majhen	droben
verjetnost	0.1	0.2	0.3	0.1

4. Dokaži parcialno pravilnost algoritma za računanje komplementa množice bitov, ki je podana kot polje:

```
const n = 100; type MNOZICA = array[1..n] of boolean;

procedure komplement(A: MNOZICA; var B : MNOZICA);
  (* fi(A,B) = (1<= i <=n : (A[i] = true) or (A[i] = false)) *)
var i : integer;
begin
  i:= n;
  while i > 0 do begin
    if A[i] = true then B[i] := false else B[i] := true;
    i := i - 1
  end; (* while *)
  (* psi(A,B) = (1<= i <=n : ((A[i] = true) and (B[i] = false))
    or ((A[i] = false) and (B[i] = true))) *)
end; (* komplement *)
```

Neobvezno (za dodatnih 10 točk): dokaži totalno pravilnost tega algoritma.