Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

10. VII 2015. godine

Diskretne i kombinatorne metode za računarsku grafiku

Dat je algoritam za sortiranje

1: **procedure** INSERTION SORT(A)
2:     **for** $j \leftarrow 2$ **to** length(A) **do**
3:         $key \leftarrow A[j]$
4:         $i \leftarrow j - 1$
5:         **while** $i > 0$ & $A[i] > key$ **do**
6:             $A[i+1] \leftarrow A[i]$
7:             $i \leftarrow i - 1$
8:         **end while**
9:         $A[i+1] \leftarrow key$
10:     **end for**
11: **end procedure**

1. Ako su vremena izvršavanja linija 2,3,4,5,6,7,9 redom $c_2, c_3, c_4, c_5, c_6, c_7, c_9$, koliko je vreme $T(n)$ izvršavanja algoritma INSERTION SORT za ulazni niz dužine n?

   1: **procedure** INSERTION SORT(A)
   2:     **for** $i \leftarrow 2$ **to** length(A) **do** $\triangleright\ n$
   3:         $key \leftarrow A[i]$ $\triangleright\ n-1$
   4:         $j \leftarrow i - 1$ $\triangleright\ n-1$
   5:         **while** $j > 0$ & $A[j] > key$ **do** $\triangleright\ \sum_{i=2}^{n} t_i$
   6:             $A[j+1] \leftarrow A[j]$ $\triangleright\ \sum_{i=2}^{n}(t_i - 1)$
   7:             $j \leftarrow j - 1$ $\triangleright\ \sum_{i=2}^{n}(t_i - 1)$
   8:         **end while**
   9:         $A[j+1] \leftarrow key$ $\triangleright\ n-1$
   10:     **end for**
   11: **end procedure**

   Vreme izvršavanja algoritma:

   $$T(n) = c_2 n + (c_3 + c_4)(n-1) + c_5 \sum_{2}^{n} t_i + (c_6 + c_7) \sum_{2}^{n} (t_i - 1) + c_9(n-1),$$

   gde je t_i broj koliko puta će i-ti element trebati da se pomeri prema početku plus jedan. t_i je najmanje 1, a najviše i puta.

2. Koji je najgori slučaj ulaznog niza za INSERTION SORT i koliko je za najgori slučaj vreme $T(n)$ iz prvog zadatka?

   Najgori slučaj ulaznog niza je obrnuto sortirani niz različitih elemenata. Tada je $t_i = i$,
   $T_W(n) = c_2 n + (c_3 + c_4)(n-1) + c_5(n(n+1)/2 - 1) + (c_6 + c_7)(n-1)n/2 + c_9(n-1) = \Theta(n^2)$

3. Koji je najbolji slučaj ulaznog niza za INSERTION SORT i koliko je za najbolji slučaj vreme $T(n)$ iz prvog zadatka?

   Najbolji slučaj ulaznog niza je sortirani niz. Tada je $t_i = 1$,
   $T_B(n) = c_2 n + (c_3 + c_4)(n-1) + c_5(n-1) + c_9(n-1) = \Theta(n)$

4. Dati definiciju velikog Θ ponašanja i pokazati po definiciji da je $n^2 - 4n\sqrt{n} = \Theta(n^2)$.

   $\Theta(g) = \{f | (\exists c_1 > 0)(\exists c_2 > 0)(\exists n_0 \in \mathbb{N})(\forall n)\ (n \geq n_0) \Rightarrow (0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n))\}$

   Umesto da pišemo $f \in \Theta(g)$, pišemo $f = \Theta(g)$ i čitamo:
   funkcija f se ponaša kao $\Theta(g)$ (kao veliko theta od g).

   $n^2 - 4n\sqrt{n} = \Theta(n^2)$. U definiciji stavljamo $c_1 = \frac{1}{5}, c_2 = 1, n_0 = 25$.

   Da li je $4n^2 + n^2 \ln n = \Theta(n^2)$? Ne.

   Da li je $4n^2 + n^2\sqrt{n} = \Theta(n^2)$? Ne.

   Da li je $4n^2 + 2^n = \Theta(n^2)$? Ne.

---

5. Za polinom $p = a_0 + a_1 x + \cdots + a_n x^n$ koeficijenti $a_0, a_1, \ldots, a_n$ su smešteni u niz `double a[]`. Promenljiva `int n` sadrži stepen polinoma p, a ako je p nula polinom, sadrži `-1`.

   U programskom jeziku C napisati proceduru `double vrednost(double x, double a[], int n)` koja vraća vrednost polinoma p izračunatu primenom Hornerove sheme.

vrednost.c

```c
double vrednost(double x, double a[], int n)
{
    int i;
    double b;
    b = a[n];
    if(n<0)
        return 0.0;
    else{
        if(n==0)
            return a[0];
    }
    for(i=n-1;i>0;i--){
        b = x * b + a[i];
    }
    return x*b + a[0];
}
```

6. Na grafu $G = (V, E)$ data je težinska funkcije $w : E \to \mathbb{R}$. Graf je zapamćen kao niz grana sa težinama. Napisato pseudo kod Kruskalovog algoritma za nalaženje minimalnog pokrivajućeg drveta.

```
function KRUSKAL(G, w)
    A ← ∅
    for each v ∈ V[G] do
        MAKE-SET(v)                         ▷ za svaki element skup koji ga sadrži
    end for
    sort(E, w)                              ▷ sortiraj grane iz E neopadajuće po w
    for each (u, v) ∈ E[G] do               ▷ redom, neopadajuće po w
        if FIND-SET(u) ≠ FIND-SET(v) then
            A ← A ∪ {(u, v)}                ▷ dodaj granu
            UNION(u, v)                     ▷ spoji skupove
        end if
    end for
    return A
end function
```

7. Da li je graf sa slike desno usmereni aciklični graf (DAG)? Obrazložiti.

   Da, ovo je usmereni graf, koji nema konturu, odnosno: aciklični.

   Ako se ignorišu usmerenja grana, da li je graf sa slike drvo? Obrazložiti.

   Ne, ima konturu: 2 - 6 - 11 - 7 - 2. I očigledno nije povezan.

   Ako je graf sa slike DAG, napisati redosled čvorova koji daje topološko sortiranje dobijeno primenom DFS algoritma.

   Izvršavanje DFS algoritma leksikografskim redosledom daje d i f vremena. Potom sortiramo čvorove opadajuće po f vremenima i dobijamo topološko sortiranje: 4, 9, 13, 3, 8, 12, 2, 7, 6, 11, 1, 5, 10, 14.

   Nije prikazan rezultat primene DFS algoritma. Postoje i drugi rezultati koji se mogu dobiti napamet.

8. Softuerska kompanija je zaposlila 5 pripravnika (A, B, C, D, E). Po jedan pripravnik će biti angažovan u 5 departmana kompanije (1, 2, 3, 4, 5). Da bi odredili koji pripravnik će se angažovati u kojem departmanu, uradili su test iz veština koje se koriste u odgovarajućem departmanu.

   U tabeli su dati poeni osvojeni na testu. Naći optimalno angažovanje.

	1	2	3	4	5
B	132	140	162	125	**128**
C	118	**142**	150	122	120
A	121	130	**160**	115	124
D	110	129	148	**117**	115
E	**130**	135	155	120	118

$\rightarrow$

30	22	0	37	34
44	20	12	40	42
41	32	2	47	38
52	33	14	45	47
32	27	7	42	44

Da bi se primenila Mađarska metoda koja traži minimum angažovanih koeficijenata preračunavaju se brojevi koliko fali od ostvarenog broja poena do maksimalnog broja: 162, od tih razlika se traži minimum.

Podvlačenjem je označeno optimalno angažovanje dobijeno Mađarskom metodom. Ukupan zbir poena dobijen optimalnim angažovanjem je 128+142+160+117+130=677.