

Úlohy na precvičenie – PDS 2019 – séria F

Úlohy riešte samostatne a podrobne. Celý postup zaznamenajte a komentujte. Odpovede zdôvodňujte celými vetami (väčšinou je otázok v úlohe viac!). Cieľom cvičenia nie je vyhľadanie riešenia, ale získanie skúseností so samostatným riešením úloh. Vo výnimočných situáciách použité citácie uveďte v riešení. Je slušné tiež uviesť mená osôb, s ktorými ste riešenie konzultovali (čo samozrejme je povolené, pokiaľ riešenie spíšete potom samostatne). Uprednostnite ručný zápis – umožní vám jednoducho pridávať ilustračné obrázky a nákresy.

Za každé správne a úplné riešenie (samozrejme aj s postupom) možno získať v tejto sérii **dva body**. Zlomky bodov možno získať aj za čiastočné riešenia. Riešenia tejto série je nutné doručiť do **4. 4. 2019, 14:25** (do začiatku tutoriálu). Pred týmto termínom je možné odovzdať riešenia na sekretariáte Ústavu informatiky (do môjho priechinka). Neskôr dodané a opisované riešenia nebudú opravované ani hodnotené. Problémy môžete konzultovať po prednáške, elektronickou poštou alebo prostredníctvom diskusnej skupiny fpds2019@googlegroups.com.

1. Zapište synchronný distribuovaný algoritmus, ktorý metódou záplavy identifikuje proces s druhým najvyšším `myid`. Každý proces na začiatku pozná svoje jednoznačné `myid` a pozná zoznam susedných procesov, s ktorými môže obojsmerne komunikovať a celkový počet procesov. Ukážte, že váš algoritmus sa skončí tak, že každý proces bude mať v premennej `subleader` hodnotu `false` okrem procesu s druhým najvyšším `myid`, ktorý bude jediný mať v tejto premennej hodnotu `true`. Riešenie komentujte a spočítajte časovú a komunikačnú zložitosť navrhnutého algoritmu.

2. Zapište synchronný distribuovaný algoritmus, ktorý umožní koordinátorovi spočítať počet procesov, ktoré sú s nim v sieti (v jej súvislom komponente). Predpokladáme, že každý proces pozná zoznam susedných vrcholov a môže s nimi obojsmerne komunikovať. Odsimulujte priebeh algoritmu (po jednotlivých distribuovaných krokoch) v sieti s procesmi 1, ..., 7 s obojsmernými komunikačnými kanálmi medzi dvojicami (12), (13), (15), (16), (23), (34), (45), (56) a (67). Koordinátorom je proces 1.

3. Uvažujme synchronný distribuovaný algoritmus, ktorý nájde najkratšiu cestu (ak ich je viac, tak práve jednu z nich) medzi zadanými procesmi p a q . Začína proces p a posiela vlnu, ktorá sa buď odrazí od listov orientovaného komunikačného podstromu, alebo nájde proces q a túto informáciu pošle späť k procesu p (na prednáške bola modifikácia, ktorou sme vyhľadávali zadaný objekt v sieti). Vzdialenosť zistí algoritmus napr. postupným počítaním návratových krokov od q k p . Predpokladajme, že proces q nie je ďaleko od p a sieť je veľmi rozľahlá. Aby sa pôvodná vlna nešírila príliš ďaleko, chceme ju zastaviť po dosiahnutí odpovede od q z procesu p vyslaním rýchlejšej vlny, ktorá zastaví pôvodnú vlnu. Sformulujte čo najpresnejšie návrh takto modifikovaného synchronného algoritmu (podľa myšlienky spomalenej vlny z prednášky) (návrh nemusíte zapisovať syntakticky presne, ale z formulácie by malo byť jasné, ako by sa riešenie naprogramovalo) a vypočítajte jeho časovú zložitosť (pre vzdialenosť d medzi procesmi p a q). Pre akú vzdialenosť d použije tento algoritmus menší celkový počet správ, ako keby sme nechali pôvodnú vlnu prejsť až k listom (pre jednoduchosť predpokladajte, že distribuovaná sieť má topológiu binárneho stromu s výškou m (má $(2^m)-1$ procesov) a proces p je v jeho koreni)?