

Zadanie 2 – Sociálna sieť

Zobudili ste sa do rána, v ktorom ste hrdým majiteľom najväčšej sociálnej siete na svete. Konkurencia vám dýcha na krk a vymysleli ste novinku, na ktorú všetci čakali: nie, nie je to nový Newsfeed, ale *modul na organizáciu súťaží*.

Hlavný komponent nového modulu je schopný efektívne vyberať fanúšikov stránky, ktorí získajú ocenenie. Na sociálnych sieťach ale „náhodný výber“ nemusí vždy dopadnúť tak, ako by ste si predstavovali: môže sa predsa ľahko stať, že aj náhodne vybraní ocenení budú ľudia s podobným menom... takú blamáž si Vaša sociálna sieť predsa nemôže dovoliť! Nezávislé médiá a trollovia, ktorí sa nerozumejú základným princípom matematickej štatistiky, by Vám nikdy neuverili, že ste to vyžrebovali náhodne.

Ako šéf ste sa preto rozhodli, že žrebovanie musí aj pôsobiť dôveryhodne a teda chcete nejako zabezpečiť, aby sa nevybrali ľudia s menami, ktoré sú abecedne príliš blízko.

Uvažujte abecedne usporiadaný zoznam fanúšikov stránky: fanúšikovia môžu priebežne pribúdať (like) alebo ubúdať (unlike). Pre potreby žrebovania, nás môže v nejakom okamihu zaujímať, kto je práve k-ty fanúšik v abecednom usporiadaní všetkých fanúšikov danej stránky.

Túto funkcionálnosť budete realizovať nasledovnými funkciami:

```
void init();
void like(char *page, char *user);
void unlike(char *page, char *user);
char *getuser(char *page, int k);
```

Vo vašom riešení implementujte tieto funkcie. Testovač najskôr zavolá funkciu `init()`, v ktorej si môžete pripraviť vaše dátové štruktúry v globálnych premenných. Následne bude volať funkcie `like`, `unlike` a `getuser`, s nasledovným významom:

```
void like(char *page, char *user)
```

Používateľ s menom `user` lajkoval stránku s názvom `page`.

```
void unlike(char *page, char *user)
```

Používateľ s menom `user` odlajkoval stránku s názvom `page`.

```
char *getuser(char *page, int k)
```

Zaujíma nás meno k-teho používateľa v abecednom poradí, ktorý lajkuje stránku `page`. Ak je týchto používateľov menej ako `k`, tak funkcia vráti `NULL`.

Vďaka tejto funkcii sa vašej sociálnej sieti začalo veľmi dariť, a prichádzajú na ňu desiatky miliónov požiadaviek za sekundu. Implementujte vyššie uvedenú funkcionálnosť čo možno najefektívnejšie.

Zadanie sa vypracúva v dvoch častiach:

- 1) Vlastný algoritmus pre sociálnu sieť (do 6.11.2017 11:59)
- 2) Dokumentácia (do 13.11.2016 11:59)

Odovzdávate program a dokumentáciu. Program je napísaný v programovacom jazyku C, zachováva určité konvencie písania prehľadných programov (pri odovzdávaní povedzte cvičiacemu, aké konvencie ste pri písaní kódu dodržiavali). Snažte sa, aby to bolo na prvý pohľad pochopiteľné.

Textová dokumentácia obsahuje hlavičku (kto, aké zadanie odovzdáva), stručný opis použitého algoritmu, s názornými nákresmi/obrázkami, a krátkymi ukážkami zdrojového kódu, vyberajte len kód na ktorý chcete extra upozorniť. Pri opise sa snažte dbať osobitý dôraz na zdôvodnenie správnosti vášho riešenia – teda, dôvody prečo je dobré/správne. Nakoniec musí technická dokumentácia obsahovať odhad výpočtovej (časovej) a priestorovej (pamäťovej) zložitosti vášho algoritmu. Celkovo musí byť cvičiacemu jasné, že viete čo ste spravili, že viete odôvodniť, že to je správne riešenie, a vedieť aké je to efektívne.

Ako budeme vaše riešenie hodnotiť?

Môžete získať 10 bodov, minimálna požiadavka 4 body.

7 bodov je za program a testovanie, 3 body sú za dokumentáciu, pričom body môžu byť výrazne ovplyvnené prezentáciou cvičiacemu (napr. keď neviete reagovať na otázky vzniká podozrenie, že to nie je vaša práca, a teda je hodnotená 0 bodov).

Dokumentácia musí obsahovať podrobný opis ako ste si svoje riešenie testovali. Ideálne, ak budete mať automatizovaný testovač, podobne ako ste mali mať v prvom zadaní. (protokoly z testovača Turing sú len ako podklad k hodnoteniu, nie ako “váš” spôsob testovania).

Prajeme veľa úspechov.