

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.

Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.

Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.

Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

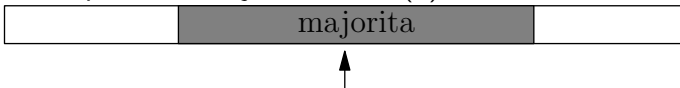
Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.
Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadať pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.



Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdeľme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.

Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Turnaj dvojíc pokračuje

Postup opakujeme až kým neskončíme s jediným prvkom.

Worst case: Zakaždým zmenšíme počet prvkov \sim na polovicu.

$$n + n/2 + n/4 + \dots = 2n$$

Časová aj pamäťová zložitosť $\Theta(n)$ ale ľahká implementácia

Optimálne riešenie

Vylepšíme predchádzajúce riešenie tak, aby stačila pamäť $O(1)$:

```
kandidat, pocet = None, 0
for p in postupnost_prvkov:
    if pocet == 0:
        kandidat, pocet = p, 1
    else:
        if p == kandidat: pocet += 1
        else: pocet -= 1
```

Úloha o majoritnom prvku a jej lepšie riešenia

Turnaj dvojíc pokračuje

Postup opakujeme až kým neskončíme s jediným prvkom.

Worst case: Zakaždým zmenšíme počet prvkov \sim na polovicu.

$$n + n/2 + n/4 + \dots = 2n$$

Časová aj pamäťová zložitosť $\Theta(n)$ ale ľahká implementácia

Optimálne riešenie

Vylepšíme predchádzajúce riešenie tak, aby stačila pamäť $O(1)$:

```
kandidat, pocet = None, 0
for p in postupnost_prvkov:
    if pocet == 0:
        kandidat, pocet = p, 1
    else:
        if p == kandidat: pocet += 1
        else: pocet -= 1
```

Všeobecnejšia verzia

Viac častých prvkov?

Na vstupe je postupnosť n prvkov a číslo k .

Nájdite všetky prvky, ktoré sa vyskytujú viac ako (n/k) -krát.

Veľmi podobné riešenie

Udržiavame množinu $k - 1$ kandidátov.

Na konci sú v nej zaručene všetky časté prvky.

(Možno aj nejaké iné, môže byť treba dodatočnú kontrolu.)

Všeobecnejšia verzia

Viac častých prvkov?

Na vstupe je postupnosť n prvkov a číslo k .

Nájdite všetky prvky, ktoré sa vyskytujú viac ako (n/k) -krát.

Veľmi podobné riešenie

Udržiavame množinu $k - 1$ kandidátov.

Na konci sú v nej zaručene všetky časté prvky.

(Možno aj nejaké iné, môže byť treba dodatočnú kontrolu.)

Praktické aplikácie a súvisiace témy

Načo je to dobré?

Lacná analýza veľkého prúdu dát:

- ktoré stránky sú najnavštevovanejšie?
- nerobí niekto DoS na môj router?

(to celé v konštantnej pomocnej pamäti)