

Lausnir á völdum dæmum úr viku þrjú

Bergur Snorrason

3. febrúar 2022

- ▶ Ég mun leysa eftirfarandi dæmi:
 - ▶ *Veci,*
 - ▶ *HKIO,*
 - ▶ *Planetaris.*

- ▶ Okkur er gefin tala $1 \leq n < 10^6$.
- ▶ Hver er minnsta talan stærri en n sem inniheldur nákvæmlega sömu tölustafi?
- ▶ Ef engin slík tala er til er svarið 0.
- ▶ 156 -> 165.
- ▶ 330 -> 0.
- ▶ 27711 -> 71127.

- ▶ Það er freystandi að reyna að útfæra gráðuga lausn.
- ▶ En þess er ekki þörf.
- ▶ Tökum eftir að svarið hefur aldrei fleiri tölustafi en inntakið.
- ▶ Svo svarið er minna en 10^6 .
- ▶ Okkur nægir því að skoða allar heiltölur á bilinu $[n + 1, 10^6]$.
- ▶ En hvernig gáum við hvort tvær tölur hafi sömu tölustafi?

Veci

- ▶ Látum x vera heiltölu.
- ▶ Þá getum við fundið aftasta tölustafinn í x með $x \% 10$.
- ▶ Við getum svo fjarlægt aftasta stafinn í x með $x / 10$.
- ▶ Við fáum því eftirfarandi.

Veci

```
3 int check(int a, int b)
4 {
5     int c[10], i;
6     for (i = 0; i < 10; i++) c[i] = 0;
7     while (a > 0) c[a%10]++, a /= 10;
8     while (b > 0) c[b%10]--, b /= 10;
9     for (i = 0; i < 10; i++) if (c[i]) return 0;
10    return 1;
11 }
```

- ▶ Við þurfum nú bara að ítra í gegnum heiltölurnar á $[n + 1, 10^6]$.

Veci

```
13 int find(int n)
14 {
15     int x = n + 1;
16     while (x < 1000000 && !check(x, n)) x++;
17     return x < 1000000 ? x : 0;
18 }
```


Veci

- ▶ Í heildina verður þetta:

Veci

```
1 #include <stdio.h>
2
3 int check(int a, int b)
4 {
5     int c[10], i;
6     for (i = 0; i < 10; i++) c[i] = 0;
7     while (a > 0) c[a%10]++, a /= 10;
8     while (b > 0) c[b%10]--, b /= 10;
9     for (i = 0; i < 10; i++) if (c[i]) return 0;
10    return 1;
11 }
12
13 int find(int n)
14 {
15     int x = n + 1;
16     while (x < 10000000 && !check(x, n)) x++;
17     return x < 10000000 ? x : 0;
18 }
19
20 int main()
21 {
22     int n;
23     scanf("%d", &n);
24     printf("%d\n", find(n));
25     return 0;
26 }
```

- ▶ Tökum eftir að við ítrum í gegnum $\mathcal{O}(n)$ tölur.
- ▶ Tímaflækjan á samanburðinum er línulegur í lengd tölustafanna, það er að segja $\mathcal{O}(\log n)$.
- ▶ Svo tímaflækjan í heildina er $\mathcal{O}(n \log n)$.

- ▶ Þetta dæmi má útfæra á eftirfarandi hátt í Python.

Veci

```
1 def check(n, x):
2     return "".join(sorted(list(str(n)))) != "".join(sorted(list(str(x))))
3
4 n = int(input())
5 x = n + 1
6 while x < 10**6 and check(n, x): x += 1
7 if x == 10**6: print(0)
8 else: print(x)
```

- ▶ Gefnar eru n heiltölur a_1, \dots, a_n .
- ▶ Finnið $j \leq k$ þannig að meðaltalið $\text{avg}(a_j, a_{j+1}, \dots, a_k)$ sé hámarkað.
- ▶ Gefið er að $1 \leq n \leq 10^5$.

- ▶ Látum m vera heiltölu þannig að $a_m = \max(a_1, \dots, a_n)$.
- ▶ Takið þá eftir að

$$\begin{aligned}\text{avg}(a_j, a_{j+1}, \dots, a_k) &= \frac{a_j + a_{j+1} + \dots + a_k}{k - j + 1} \\ &\leq \frac{a_m + a_m + \dots + a_m}{k - j + 1} \\ &= \frac{(k - j + 1) \cdot a_m}{k - j + 1} \\ &= a_m.\end{aligned}$$

- ▶ Svo meðaltalið verður aldrei stærra en a_m .
- ▶ En einnig gildir að $\text{avg}(a_m) = a_m$.
- ▶ Svo okkur nægir að finna stærstu töluna í listanum.

HKIO

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i, n, r;
6     scanf("%d", &n);
7     int a[n];
8     for (i = 0; i < n; i++) scanf("%d", &a[i]);
9     for (r = i = 0; i < n; i++) if (a[i] > a[r]) r = i;
10    printf("%d %d\n", r, r);
11    return 0;
12 }
```


Planetaris

- ▶ Atli og Finnur eru að spila tölvuleik sem snýst um að fanga sólkerfi.
- ▶ Í leiknum eru $1 \leq n \leq 10^5$ sólkerfi.
- ▶ Atli og Finnur senda einhvern fjölda skipa sinna á hvert sólkerfi.
- ▶ Atli fangar tiltekið sólkerfi ef hann sendir strangt fleiri skip þangað.
- ▶ Atli hefur a skip og veit að Finnur mun senda e_i skip á i -ta sólkerfið.
- ▶ Hver er mesti fjöldi sólkerfa sem Atli getur fangað?

Planetaris

- ▶ Við græðum jafn mikið að fanga hvert sólkerfi, svo það er best að fanga þau sólkerfi sem Finnur sendir fá skip á.
- ▶ Við föngum því einfaldlega sólkerfin í röð, byrjum á því sem Finnur sendir fæst skip á, svo næst það sem hann sendir næst fæst skip á, og svo framvegis.
- ▶ Þegar við föngum i -ta sólkerfið verðum við að passa að senda $e_i + 1$ skip, til að það verði ekki jafntefli.
- ▶ Við verðum líka að passa að hætta að fanga sólkerfi þegar við höfum ekki nóg af skipum.

Planetaris

```
10 int main()
11 {
12     int i, n, a, e[MAXN];
13     scanf("%d%d", &n, &a);
14     for (i = 0; i < n; i++) scanf("%d", &e[i]);
15     qsort(e, n, sizeof *e, cmp);
16     for (i = 0; i < n; a -= e[i++] + 1) if (a < e[i] + 1) break;
17     printf("%d\n", i);
18     return 0;
19 }
```

Planetaris

- ▶ Tímaflækjan á þessari lausn er $\mathcal{O}(n \log n)$ sökum röðunar.
- ▶ Takið eftir að það er mjög auðvelt að gera litlar villur sem gera lausnin ranga.
- ▶ Til dæmis fær eftirfarandi lausn rétt í sýnidæmum en rangt á fyrsta huldudæminu.

Planetaris, röntg lausn

```
10 int main()
11 {
12     int i, n, a, e[MAXN];
13     scanf("%d%d", &n, &a);
14     for (i = 0; i < n; i++) scanf("%d", &e[i]);
15     qsort(e, n, sizeof *e, cmp);
16     for (i = 0; i < n; i++)
17     {
18         a -= e[i] + 1;
19         if (a <= e[i] + 1) break;
20     }
21     printf("%d\n", i + 1);
22     return 0;
23 }
```

