Programiranje 2 — izpit 1
11. junij 2024

Vsa števila, ki nastopajo v besedilu in testnih primerih, so cela.

Razdelitev točk po nalogah: 30, 35, 35.

Oddajte datoteke `naloga1.c`, `naloga2.c` in `naloga3.c`.

1. V prvi vrstici vhodne datoteke sta podani števili $n \in [3, 9]$ in $p \in [1, 100]$, nato pa sledi p vrstic, od katerih vsaka vsebuje zaporedje najmanj dveh in največ $2n^2$ nizov dolžine 2, ločenih s po enim presledkom. Vsak niz predstavlja eno od polj na šahovnici velikosti $n \times n$. Prvi znak niza predstavlja oznako stolpca (`a` je prvi stolpec, `b` drugi itd.), drugi znak pa zaporedno številko vrstice (ta se začne z 1).

Napišite program, ki v podano izhodno datoteko izpiše p vrstic. Če i-to zaporedje nizov v vhodni datoteki predstavlja veljaven skakačev sprehod po šahovnici in če vsako polje šahovnice v tem zaporedju nastopa natanko enkrat (upoštevamo tudi začetno polje sprehoda), naj bo v i-ti vrstici izhodne datoteke zapisano število 1, sicer pa število 0. Imeni vhodne in izhodne datoteke sta podani kot argumenta ukazne vrstice.

Skakač se s polja (v, s) lahko premakne na vsa polja (v', s'), za katera velja $|v' - v| \cdot |s' - s| = 2$.

Sledeči primer prikazuje vsebino datoteke `vhod01.txt` in datoteke `rezultat01.txt`, ki naj bi jo program ustvaril po izvedbi naslednjih ukazov:

```
gcc -o naloga1 naloga1.c
./naloga1 vhod01.txt rezultat01.txt
```

`vhod01.txt`:

```
5 4
c3 e2 d4 b5 a3 b1 d2 e4 c5 a4 b2 d1 e3 d5 b4 a2 c1 d3 e1 c2 a1 b3 a5 c4 e5
c3 e2 d4 b5 a3 b1 d2 e4 c5 a4 b2 d1 e3 d5 b4 a2 c1 d3 e1 c2 a1 b3 a5 c4 e5 d3
c3 e2 d4 b5 a3 b1 d2 e4 c5 a4 b2 c4 e3 d5 b4 a2 c1 d3 e1 c2 a1 b3 a5 c4 e5
c3 e2 d4 b5 a3 b1 d2 e3 c5 a4 b2 d1 e4 d5 b4 a2 c1 d3 e1 c2 a1 b3 a5 c4 e5
```

`rezultat01.txt`:

```
1
0
0
0
```

V drugem zaporedju se polje `d3` ponovi, tretje zaporedje ne pokrije celotne šahovnice, četrto pa ne predstavlja veljavnega skakačevega sprehoda.

V 50% testnih primerov je število nizov v vsaki vrstici vhodne datoteke enako natanko n^2.

(2) V datoteki `naloga2.h` je podana definicija strukture `Vozlisce`, ki predstavlja vozlišče povezanega seznama. V datoteki `naloga2.c` dopolnite funkcijo

```
void obdelaj(Vozlisce* zacetek, int k),
```

tako da bo v smeri od začetka do konca povezanega seznama odstranila vsa vozlišča, pri katerih je vrednost komponente `podatek` enaka vsoti komponent `podatek` predhodnih k neodstranjenih vozlišč seznama. Prvo vozlišče seznama se nahaja na naslovu `zacetek`.

Na primer, če je $k = 3$, vozlišča seznama pa hranijo podatke 5, 4, 9, 18, 18, 7, 20, 6, 0, 13, 19, 2 in 21, naj funkcija izbriše obe vozlišči s podatkom 18 ($= 5 + 4 + 9$), vozlišče s podatkom 20 ($= 4 + 9 + 7$), vozlišče s podatkom 13 ($= 7 + 6 + 0$) in vozlišče s podatkom 21 ($= 0 + 19 + 2$).

Naj bo n število vozlišč seznama, D pa minimalna razdalja med zaporednima vozliščema, ki ju je treba odstraniti. V vseh testnih primerih velja $n \in [1, 2 \cdot 10^5]$ in $k \in [1, n]$, poleg tega pa velja še naslednje:

- V 20% testnih primerov je $n \leqslant 2000$, $k = 1$ in $D \geqslant k + 1$.
- V nadaljnjih 20% testnih primerov je $n \leqslant 2000$ in $D \geqslant k + 1$.
- V nadaljnjih 40% testnih primerov je $n \leqslant 2000$.

(3) Pri problemu hanojskih stolpov želimo n kolutov medsebojno različnih premerov, ki so po padajočih premerih nataknjeni na palico A, s pomočjo pomožne palice B prestaviti na palico C, ne da bi kadarkoli položili večji kolut na manjši. Optimalni postopek reševanja je enolično določen in zahteva $2^n - 1$ potez. Napišite program (`naloga3.c`), ki s standardnega vhoda prebere števili $n \in [1, 20]$ in $k \in [0, 2^n - 1]$, na standardni izhod pa izpiše zaporedne številke potez v optimalnem postopku, v katerih prestavimo najmanjši kolut od koderkoli na palico B, pri čemer prvih k potez postopka ignoriramo.

Sledi primer:

`test01.in`:

```
4 5
```

`test01.out`:

```
7
13
```

Optimalno zaporedje potez je $A \to B$, $A \to C$, $B \to C$, $A \to B$, $C \to A$, $C \to B$, $A \to B$, $A \to C$, $B \to C$, $B \to A$, $C \to A$, $B \to C$, $A \to B$, $A \to C$, $B \to C$. Najmanjši kolut prestavimo na palico B v prvi, sedmi in trinajsti potezi. Ker je $k = 5$, prvih pet potez ignoriramo.