

# Palindrom

---

Mladi Huan je, radeći zadaću iz kontinualne matematike, naletio na sljedeći jako zanimljiv zadatak čije je rješenje trivijalno za sve studente koji redovno pohađaju tutorijale i predavanja. Nažalost, Huan nije takav student pa je odlučio unajmiti vas da mu 'iskodirate' rješenje kako ne bi izgubio uslov za pristupanje usmenom ispitu. Zadatak je sljedeći:

U mreži gradova (čiji su indeksi cijeli brojevi) države Ang, postoji cesta između dva grada označenih sa cjelobrojnim indeksima  $x$  i  $y$  akko je  $x < y$ . Svaka cesta koja povezuje dva grada  $x$  i  $y$  je označena sa jednim velikim slovom engleskog alfabeta. Da biste došli iz grada  $a$  u grad  $b$ , potrebno je držati se zakona koji kaže da konkatencija svih slova nađenih na oznakama svake ceste na putu od grada  $a$  do grada  $b$  mora sačinjavati palindrom. Država Ang, kao mjeru kompleksnosti putne mreže, uzima dužinu najvećeg puta koji je moguće ostvariti od grada  $0$  do grada  $n - 1$ . Pošto je ova statistika od vitalne važnosti za cijelo stanovništvo, potrebno je pronaći najduži palindrom koji je moguće konstruisati putovanjem iz grada sa indeksom  $0$  do grada sa indeksom  $n - 1$ . Ako postoji više rješenja, potrebno je pronaći ono koje je leksikografski najmanje.

Funkcija koju trebate implementirati kako bi riješili ovaj zadatak ima jedan parametar. Taj parametar je vektor "parova parova", pri čemu prvi član prvog par predstavlja 2 grada (podsjećamo, gradovi su označeni cjelobrojnim indeksima), a drugi član prvog para predstavlja slovo koje označava tu cestu (pogledati pojašnjenje primjera).

```
std::string palindrom(const std::vector<std::pair<std::pair<int, int>, char>> &G); [C++]
```

Gdje parametri redom predstavljaju:

**G** - mreža gradova opisana u prethodnom tekstu.

**Povratni tip std::string** - rješenje zadatka.

**Napomena:** indeks  $n - 1$  (opisan u tekstu zadatka) će uvijek biti najveći indeks u ulaznom vektoru.

```
char* palindrom(int n, int *a, int *b, char *c); [C]
```

Funkcija palindrom je nešto drugačijeg oblika u odnosu na njenu C++ varijantu. Umjesto vektora sa elementima tipa `std::pair`, taj isti vektor je razložen na tri niza pri čemu prvi niz predstavlja grad iz kojeg se dolazi, drugi niz predstavlja grad u koji se ulazi, dok treći niz predstavlja oznaku puta između tih gradova. Parametar  $n$  predstavlja dužinu ova tri niza.

**Primjer (C++)**

Ulazni parametri	Povratna varijabla
palindrom({{0, 2}, B}, {{0, 1}, A}, {{0, 3}, A}, {{0, 4}, C}, {{1, 2}, C}, {{1, 3}, B}, {{1, 4}, D}, {{2, 3}, C}, {{2, 4}, B}, {{3, 4}, A})	"ACCA"

**Primjer (C)**

Ulazni parametri	Povratna varijabla
palindrom({0, 0, 0, 0, 1, 1, 1, 2, 2, 3}, {2, 1, 3, 4, 2, 3, 4, 3, 4, 4}, {B, A, A, C, C, B, D, C, B, C})	"ACCA"

**Pojašnjenje primjera**

Putevi države Ang su opisani vektorom sa sljedećim članovima:

{{0, 2}, B} -> (postoji cesta između grada 0 i 2 pri čemu je cesta označena sa B)

{{0, 1}, A} -> (postoji cesta između grada 0 i 1 pri čemu je cesta označena sa A)

...

{{3, 4}, A} -> (postoji cesta između grada 3 i 4 pri čemu je cesta označena sa A)

Najveći indeks u ulaznom skupu je 4, pa je  $n - 1 = 4$ . Najduži palindrom koji je moguće konstruisati putovanjem iz grada 0 u grad 4 je ACCA.

**Ograničenja na resurse i opis subtaskova**

Zadatak će biti testiran na tri podzadatka, od kojih svaki nosi određeni broj bodova i ima sljedeća ograničenja:

Neka je  $n$  broj gradova u državi, pri čemu je indeks grada uvijek između 0 i  $n - 1$ .

**Podzadatak 1 (15 bodova):**  $n \leq 10$

**Podzadatak 2 (15 bodova):**  $n \leq 20$

**Podzadatak 3 (70 bodova):**  $n \leq 200$

Vremenska i memorijska ograničenja su dostupna na sistemu za ocjenjivanje.