

Struktura podataka Trie

<https://arena.petlja.org/sr-Latn-RS/competition/2019kiaacas02>

1. Kreirati strukturu za predstavljanje čvora prefiksnog stabla. Napisati funkcije za umetanje reči u stablo i proveru da li se data reč nalazi u stablu (trie).
2. Konstruisati algoritam koji za datih n reči pronalazi najduži zajednički prefiks.
Primer: Za reči **code**, **codecs**, **coder**, **coding** najduži zajednički prefiks je *cod*.
3. Neka je dat skup reči S . Koristeći prefiksna stabla, sorirati reči leksikografski u rastućem poretku. *Za domaći:* Modifikovati algoritam tako da reči budu sortirane opadajuće.
4. U datom skupu S sa velikim brojem reči od kojih mnoge nisu unikati pronaći reč koja se javlja najveći broj puta. Ako dve reči imaju isti najveći broj pojavljivanja, ispisati ma koju. Za rešenje zadatka koristiti prefiksna stabla.
5. Koristeći prefiksna stabla, pronaći i ispisati k reči koje se javljaju najveći broj puta u datom skupu reči S .
6. Neka je data binarna matrica (svi elementi su ili 0 ili 1). Koristeći prefiksna stabla pronaći i ispisati redne brojeve svih vrsta koje se javljaju kao duplikati. Ukoliko su vrste 3 i 5 iste ispisati samo 5. *Za domaći:* Kako se ovaj problem može modifikovati tako da ispisuje i 3 i 5 ako su one duplikati?
7. Neka je dato n binarnih brojeva. Koristeći prefiksna stabla pronaći maksimalni broj koji se može dobiti XOR-ovanjem (ekskluzivna disjunkcija) ovih brojeva. *Za domaći:* Modifikovati algoritam tako da nalazi najmanji XOR.

8. Neka je dat skup reči D i neka niska s (bez belina). Proveriti da li se pomoću reči iz skupa D može formirati niska s .

Primer: $D = \{\text{"word", "br", "k", "b", "ea", "pr", "oblem"}\}$,
 $s = \text{"wordbreakproblem"}$. Pomoću "word", "br", "ea", "k", "pr" i "oblem" se može formirati niska s pa je u ovom slučaju odgovor na pitanje da li se niska s može izdeliti na reči iz skupa D da. Da je skup D bio $D = \{\text{"word", "br", "b", "ea", "pr", "oblem"}\}$ odgovor bi bio negativan jer nam fali slovo k .

Za domaći, **ko želi:** Koristeći prefiskna stabla implementirati auto-complete funkciju. Napraviti rečnik podržanih reči, zatim sa standardnog ulaza uneti početak reči, a kao rezultat ispisati sve one reči koje se mogu dobiti dopunjavanjem unetog prefiksa.

Svi domaći su neobavezni i ne ocenjuju se. Služe za utvrđivanje gradiva i one koji žele da provežbaju predjeno gradivo sa časa.