1. Sestavi predstavitev vrste v programskem jeziku Python tako, da boš za hranjenje elementov uporabil verižni seznam, vse osnovne operacije pa bodo izvedljive v časovni zahtevnosti O(1).
2. Sestavi algoritem, ki določi najdaljšo pot od korena do lista. Če je v drevesu le koren, je ta pot dolga 0, če je drevo prazno, gre za napako, za drevo



je ta pot dolžine 5 (do lista H4 ali H5).

1. Dano je iskalno dvojiško drevo, v katerem hranimo števila. Sestavi algoritem, ki izpiše vse podatke iz tega drevesa padajoče, torej denimo 20, 10, 5, 4, 2, 1!
2. V podatkovni strukturi vrsta kot podatke lahko hranimo **cela števila**. Imamo vrsto, kjer so vsi podatki **pozitivni**. Sestavi algoritem **prostorske zahtevnosti O(1),** ki to vrsto spremeni tako, da bodo v njej samo še liha števila v istem relativnem vrstnem redu kot prej. Postopek vrsto 12, 7, 4, 5, 32, 1, 1, 7 torej spremeni v 7, 5, 1, 1.
3. Problem trgovskega potnika v omrežju, podanem s sledečo matriko

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| */* | *2* | *9* | *3* | *8* |
| *5* | */* | *3* | *8* | *6* |
| *4* | *6* | */* | *3* | *8* |
| *3* | *7* | *3* | */* | *6* |
| *8* | *1* | *3* | *7* | */* |

rešujemo z metodo razveji in omeji, kjer kot ocenjevalno funkcijo uporabimo redukcijo matrike. Zapiši oceno vrednosti vseh krožnih poti, ki vsebujejo povezavo 1 – 2. Zapiši tudi reducirano matriko, ki jo dobimo pri reševanju in pripada vozlišču, ki opisuje vse krožne poti, ki vsebujejo pot 1 – 4 – 2.