Predpostavka: Če imamo prazno drevo, vrnemo None

Algoritem bo deloval rekurzivno. Funkcija, ki jo bomo klicali rekurzivno poimenujemo **vsota**.

Algoritem:

1. Najprej preverimo ali je drevo prazno. Če je prazno vrnemo None.
2. Preverimo ali ima koren le levo poddrevo. V primeru, da ima le levega naredimo rekurzivni klic na levo poddrevo (vsota(levo poddrevo). Če dobimo kot rezultat None vrnemo vrednost korena, saj če smo dobili None pomeni, da je levo poddrevo prazno in ne prispeva nič k vsoti. V primeru ko ne dobimo None, vrnemo vrednost, ki smo jo dobili plus vrednost korena. Simetrično naredimo, če dobimo le desno poddrevo.
3. Ko imamo levo in desno poddrevo naredimo rekurzivni klic tako za levo kot za desno poddrevo in si zapomnimo rezultat. V primeru ko vsota(levo poddrevo) in vsota(desno poddrevo) vrne None, vrnemo None (saj to pomeni, da sta obe poddrevesi prazni). V primeru ko vsota(levo poddrevo) vrne None, vrnemo vsota(desno poddrevo), saj se očitno v desnem poddrevesu nahajajo neki koreni, ki imajo le enega sina sicer bi dobili None . V primeru ko vsota(desno poddrevo) vrne None, vrnemo vsota(levo poddrevo). Sicer pa vrnemo vsota(desno poddrevo) plus vsota(levo poddrevo).

Koda:

Slika, ki vsebuje besede besedilo, posnetek zaslona, programska oprema

Opis je samodejno ustvarjen