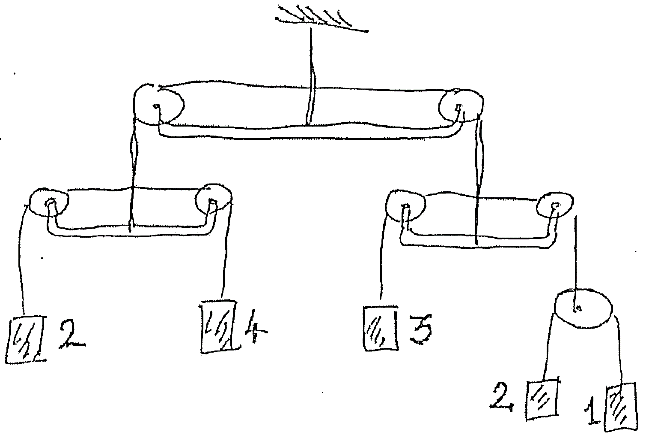
**Računalništvo 1 – praktična matematika**

**1. pisni izpit, 2.2.2017**

*Naloge so enakovredne (po 15T). Čas reševanja je 90 minut.*

1. Dano je dvojiško drevo s podatki v listih, ki predstavljajo uteži z znanimi masami, vsako notranje vozlišče pa predstavljaja škripec, s katerega visi vrv na levo in na desno stran. Na vsaki strani vrvice je bodisi utež bodisi drug škripec.



Sestavi algoritem časovne zahtevnosti , kjer je število uteži, ki preveri, ali sistem škripcev in uteži miruje, če ga obesimo pod strop. Utež sama seveda miruje, škripec pa miruje, če je vsota vseh mas na levi enaka vsoti vseh mas na desni. Cel sistem miruje, če miruje vsak škripec v sistemu. Težo vrvic in škripcev zanemarimo. Algoritem podrobno opiši in utemelji, da je res take časovne zahtevnosti.

1. Datoteka HTML je znakovna datoteka, v kateri strukturo in obliko besedila določimo s pomočjo značk. Krepko besedilo npr. opišemo z **<b>besedilo</b>**, kjer je **<b>** začetna značka z imenom b, **</b>** pa pripadajoča končna značka. Pri začetnih značkah lahko navedemo še nekatere lastnosti, kar pa za naš problem ne bo pomembno. Nekatere začetne značke so brez ustreznih zaključnih značk – v tem primeru dodamo znak **/** pri koncu začetne značke, npr. **<br />**, in rečemo, da je to začetna in končna značka hkrati.

V Pythonu je dan generator **znacke(ime\_datoteke)**, ki generira značke po vrsti, kot se nahajajo v datoteki z danim imenom. Za vsako značko dobimo nabor **(ime, tip)**, kjer je **ime** ime značke (niz),   
**tip** pa določa vrsto značke (1 = začetna značka, 2 = končna značka, 3 = začetna in končna značka).

Primer: za datoteko HTML z vsebino: **<h1>Primer <i>datoteke</i> <img src="slika.jpg" /></h1>**bi generator po vrsti vračal nabore: **('h1', 1), ('i', 1), ('i', 2), ('img', 3), ('h1', 2)**

Sestavi funkcijo, ki za datoteko z danim imenom preveri, ali so značke na njej pravilno gnezdene.

1. Reši problem 0/1 nahrbtnika za naslednje podatke:

Napiši tako optimalno vrednost kot tudi vse polnitve, s katerimi to optimalno vrednost dosežemo. Utemelji, da si res navedel vse polnitve.

1. Dan je seznam, ki vsebuje premi pregled dvojiškega drevesa. Sestavi algoritem, ki ugotovi, če je to drevo lahko iskalno. Npr. za sezname [8, 4, 2, 1, 6, 5, 10, 9, 30, 11], [1, 2, 3], [8, 10, 9, 30, 11], [3, 2, 1], [3, 1, 2], vrne True, za sezname [8, 4, 10, 2], [8, 4, 2, 1, 6, 5, 10, 7, 30, 11], [1, 2, 4, 1] pa False.