Kedy na Jedličku?

Každý študent vie, že začiatok nového semestra treba poriadne osláviť a najlepšie miesto na to v Košiciach je Jedlička (toto zadanie nebolo financované organizátormi Jedličky). Na druhej strane, ale všetci ste poctiví a zodpovední študenti, a viete že deň po Jedličke musíte byť na dôležitých prednáškach a cvičeniach, a práve preto ste sa rozhodli že na Jedličku prídete iba na jednu hodinu. Posledná otázka, ktorá zostáva, je kedy prísť na túto párty.

Pre efektívne riešenie tejto dilemy ste sa opýtali vašich priateľov, kedy oni pôjdu na Jedličku, a keďže aj oni sú poctiví a zodpovední, zistili ste, že každý z nich bude na párty iba niekoľko hodín. Vašou úlohou je na základe zoznamu časov kedy vaši priatelia budú na Jedličke vybrať si najvhodnejšiu hodinu, kedy stretnete najviac vašich priateľov. Z doposiaľ neznámych dôvodov ste sa rozhodli, že na riešenie problému implementujete jednoduchý (naozaj) Python program.

Vstupom algoritmu je zoznam intervalov, v ktorých vaši priatelia budú na Jedličke. Každý interval je reprezentovaný dvoma číslami `start` a `end`, pričom interval považujeme za zatvorený zľava a otvorený sprava: `<start, end)`. To znamená, že ak na Jedličku prídete v čase `start`, určite stretnete daného priateľa, ale dotyčný v čase `end` na párty už nebude. (Švajčiarska vláda podporila každého z vašich spolužiakov hodinkami aby ste si mohli byť istí, že dané intervaly dodržiavajú so sto percentnou istotou.)

Ako príklad zoberme zoznam:

Meno	**Príde**	**Odíde**
Maťo	9	10
Tomáš	10	12
Miro	10	1
Katka	11	1
Paľo	8	11
Dano	10	11
Mara	12	2

V tomto prípade sa vám najviac oplatí prísť na 10, keď sa stretnete s Tomášom, Mirom, Paľom a Danom.

Úloha 1 – 1 bod

Implementujte funkciu `loadIntervals`, ktorá má jeden parameter, cestu k súboru, ktorý obsahuje intervaly návštev Jedličky. Každý riadok obsahuje tri hodnoty oddelené čiarkou, kde prvá hodnota je meno, druhá hodnota celé číslo vyjadrujúce čas príchodu na párty, a posledná hodnota čas odchodu. Napríklad tabuľka uvedená vyššie by bola reprezentovaná nasledovne:

```
Maťo,9,10
Tomáš,10,12
Miro,10,1
```

```
Katka,11,1
Paľo,8,11
Dano,10,11
Mara,12,2
```

Ďalšie príklady nájdete v priečinku `schedules`.

Funkcia `loadIntervals` vracia jednu hodnotu, zoznam intervalov reprezentovaných ako n-tica (je to na vás, ktoré hodnoty použijete).

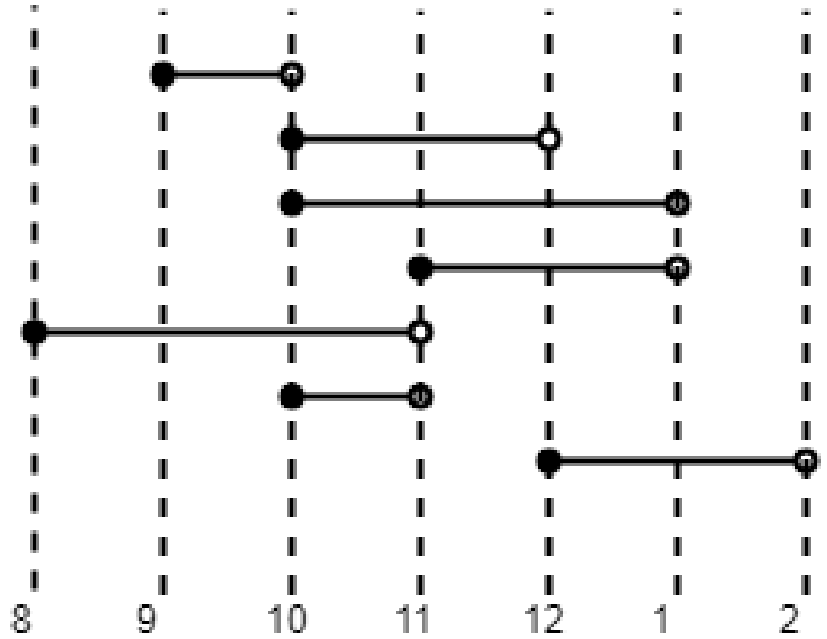
Pozor! Pri hodnotení vášho riešenia sa bude kontrolovať aj schopnosť funkcie `loadIntervals` spracovať nesprávne štruktúrované súbory – t. j. funkcia by mala upozorniť používateľa na nesprávnosť údajov v súbore.

Pomôcka
Pre jednoduchšiu prácu s načítanými údajmi je odporúčané, aby ste časy po polnoci premenili do podoby ako keby boli poobedňajšie, napr.: 1 → 13, 2 → 14, atď.

Úloha 2 – 1 bod

Implementujte funkciu `chooseTime`, ktorá má jeden parameter (zoznam n-tíc reprezentujúce intervaly návštev) a vráti dve hodnoty: najvhodnejší čas na návštevu Jedličky (hodina) a počet prítomných priateľov. V prípade príkladu uvedeného vyššie by funkcia mala vracať hodnoty `10` a `4`.

Algoritmické riešenie je na vás, ale ako často pri riešení programátorskej úlohy, vhodná reprezentácia vám uľahčí riešenie.



Napr. pri tejto reprezentácii hľadáte zvislú čiaru s najväčším počtom plných krúžkov a priesekov.

Implementujte funkciu `whenToGo`, ktorá bude slúžiť ako hlavná funkcia pri riešení úlohy. Funkcia zatiaľ má iba jeden parameter, cestu k súboru s intervalmi. Funkcia načíta intervaly zo súboru, a následne zavolá funkciu `chooseTime`. Na konci volania funkcia vypíše správu vo forme:

```
The best time to attend Jedlicka is at 10 o'clock when you'll meet 4 of your friends.
```

Pozor! Ak pri riešení úlohy 1 ste transformovali časy na popoludňajšie hodiny, nezabudnite ich znova prekonvertovať do pôvodného stavu (13 → 1).

Úloha 3 – 1 bod

Implementujte funkciu `chooseTimeConstrained`, ktorá má tri parametre: zoznam intervalov, najskoršia a najneskoršia hodina kedy môžete ísť na Jedličku. To znamená že pri volaní `chooseTimeConstrained(interval_list, 9, 11)` funkcia hľadá iba možnosti z intervalu `<9, 11)` (podobne ako pri intervaloch vašich priateľov).

Rozšírte funkciu `whenToGo` o dva parametre, `ystart` a `yend` ktoré reprezentujú interval ktorý vám vyhovuje na návštevu Jedličky. Nastavte týmto parametrom defaultné hodnoty. Ak pri volaní sa nespresní žiadny interval, zavolá sa funkcia `chooseTime`, ak používateľ zadá parametre `ystart` a `yend`, zavolá sa funkcia `chooseTimeConstrained`.

Úloha 4 – 1 bod

V kostre projekte nájdete priečinok `schedules_weights`. Súbory v tomto priečinku majú upravenú štruktúru oproti predošlým súborom. Každý riadok je rozšírený o jednu celočíselnú hodnotu, ktorá vyjadruje váhu, resp. mieru toho ako veľmi sa chcete stretnúť s daným priateľom.

```
Maťo,9,10,3
Tomáš,10,12,2
Miro,10,1,0
Katka,11,1,1
Paľo,8,11,2
Dano,10,11,1
Mara,12,2,5
```

Upravte funkciu `loadIntervals` tak, aby vedela načítať súbory oboch typov (s tromi alebo štyrmi hodnotami po riadok). Návratová hodnota funkcie sa nemení, iba n-tice budú obsahovať o jednu hodnotu viac (váha).

Implementujte funkciu `chooseTimeWithWeights`, ktorá určí najvhodnejší čas na párty nie vzhľadom na čas ale vzhľadom na súčet váh prítomných priateľov. V hornom prípade je najvhodnejšie ísť o 11, keď prítomní priatelia majú celkovú preferenciu 13. Funkcia vracia dve hodnoty: najvhodnejší čas návštevy a celkovú váhu (v našom prípade `11` a `13`).

Upravte funkciu `whenToGo`, aby – ak n-tice v zozname intervalu obsahujú aj váhu – zavolala funkciu `chooseTimeWithWeights`. Takisto upravte výpis, aby funkcia vypísala sumu váh a nie počet priateľov:

```
The best time to attend Jedlicka is at 11 o'clock when you'll meet friends with a weight of 13.
```