Instructions for translators

1. Open this file on GitHub server. If you see `https://um.mendelu.cz/...` in URL, click `View on GitHub` to open this file on github.com.
2. If you see this file on GitHub server, you can edit the content of the file. Open the file in an editor. You can use simple editor (pres e on GitHub). However, an advanced VS Code editor (press `.` on GitHub) is better, since it provides preview how the Markdown code renders. Alternatively press pencil for simple editor or press triangle next to the pencil to get access to VS Code described as `github.dev`.
3. Fix the keywords in the preamble.
4. Depending on which language version you want to use as a source for your translation, delete either English or Czech version below.
5. Translate to your language. Keep Markdown marking and math notation. If you use a tool to get first version of the translation, make sure that the markup is preserved.
6. In VS Code you can open the preview in another window by pressing `Ctrl+V` and `K`. Keep the preview open as you work, or close using a mouse.
7. Instead of saving, you have to commit and push the changes to the repository. Fill the Message under `Source control` (describe your changes, such as "Polish translation started") and then press Commit&Push.
8. Make sure that your changes appear in the commit history. In rare cases (if you work with simultaneously with someone else) you have to download /Pull/ and merge his and yours changes. Usualy Sync (Pull & Push) should work.
9. When you finish the translation, change `is_finished: False` in header to `is_finished: True`.

Instrukce pro překladatele

1. Otevřete tento soubor na serveru GitHub. Pokud máte soubor otevřen na `https://um.mendelu.cz/...`, otevřete jej na serveru github.com.
2. Pokud tento soubor vidíte na serveru GitHub, můžete obsah souboru upravit. Otevřete soubor v editoru. Můžete použít jednoduchý editor (stiskněte e na GitHubu). Lepší je však pokročilý editor VS Code (stikněte `.` na GitHubu), protože poskytuje náhled, jak se kód Markdown interpretuje. Případně stiskněte tužku pro jednoduchý editor nebo stiskněte trojúhelníček vedle tužky, abyste získali přístup k editoru VS Code popsaný jako `github.dev`.
3. Opravte klíčová slova v preambuli.
4. V závislosti na tom, kterou jazykovou verzi chcete použít jako zdrojový kód pro svůj překladu, odstraňte níže uvedenou anglickou nebo českou verzi.
5. Přeložte do svého jazyka. Ponechte značení Markdown a matematický zápis. Pokud použijete nástroj typu DeepL pro získání první verze překladu, ujistěte se, že zápis matematických výrazů byl zachován.
6. Ve VS Code můžete náhled otevřít v jiném okně stisknutím `Ctrl+V.` a `K`. Během práce nechte náhled otevřený nebo jej zavřete pomocí myši.
7. Místo uložení musíte změny zaregistrovat a odeslat do úložiště. Vyplňte zprávu v poli `Zpráva` (popište své změny, např. "Zahájen překlad do polštiny") a poté stiskněte tlačítko Commit&Push.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by the European Union

8. Ujistěte se, že se vaše změny objeví v historii revizí. Ve výjimečných případech (pokud pracujete současně s někým jiným) musíte stáhnout /Pull/ a sloučit jeho a vaše změny. Obvykle by synchronizace (Pull & Push) měla fungovat.
9. Po dokončení překladu změňte `is_finished: False` v záhlaví na `is_finished: True`.

Czech source

Round robin

Keywords: combinatorics, probability and statistics, combinatorics

Představte si, že pořádáte školní turnaj ve stolním tenise, šachu, e-sportu nebo třeba futsale. Chcete, aby byl co nejspravedlivější – aby každý hráč měl možnost utkat se se všemi ostatními. Právě k tomu slouží systém každý s každým, známý také jako round robin.

Jeho hlavní výhodou je férovost: výsledné pořadí závisí jen na výkonech hráčů nebo týmů, ne na náhodném losu soupeřů. Na druhou stranu, počet zápasů rychle roste s počtem účastníků – naplánovat takový turnaj může být docela výzva. A právě zde přichází ke slovu kombinatorika – matematika počítání možností.

Futsalový turnaj

Úloha 1. Na turnaj ve futsalu se přihlásilo 9 týmů. Je vedený systémem round robin, tzn. každý tým hraje s každým jeden zápas. Tým za každou výhru v zápase dostává 2 body, za remízu 1 bod a za prohru 0 bodů. O celkovém umístění týmu rozhodne závěrečný součet bodů za všechny zápasy
Kolik zápasů je nutné na turnaji odehrát? Kolika způsoby je možné sestavit rozvrh turnaje, je-li k dispozici jediné hřiště, na kterém se zápasy postupně odehrávají?

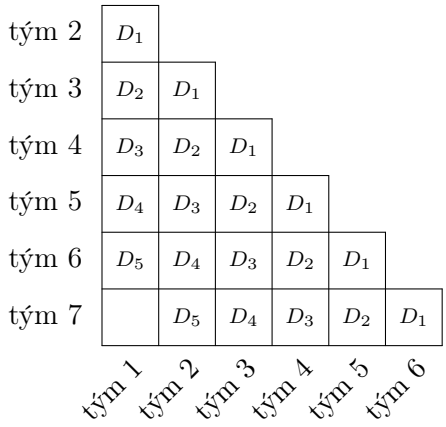
Úloha 2. Ukažte, že jestliže některý tým v turnaji z předchozí úlohy získal celkem 13 bodů, pak nutně patří mezi čtyři nejlepší týmy turnaje.

Férovější soutěž

Na další ročník futsalového turnaje z předchozích úloh se tentokrát přihlásilo 7 týmů. Při sestavování rozvrhu turnaje si však organizátor dal novou podmínku, že žádný tým nesmí hrát ve dvou zápasech těsně za sebou, aby hráči nemuseli hrát unavení a turnaj byl férovější.

Libor vymyslel algoritmus, jak požadovanou posloupnost zápasů sestavit. Vychází z následující tabulky.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by the European Union



Obr. 1: Tabulka pro tvorbu programu férového turnaje

Její každé pole v i-tém řádku a j-tém sloupci odpovídá zápasu $(i+1)$-tého a j-tého týmu. Hledaná posloupnost zápasů bude odpovídat pořadí polí, která Libor postupně vybírá. Pro přehlednost budeme tato pole značit dle týmů, jejichž zápas reprezentuje, tj. $[1;2]$, $[3;5]$ atd. Dále označíme nejdelší diagonálu začínající polem $[1;2]$ a končící polem $[6;7]$ jako D_1, kratší diagonálu začínající polem $[1;3]$ a končící polem $[5;7]$ jako D_2 apod.

Liborův algoritmus vypadá následovně: - jako první vybereme pole v prvním sloupci a posledním řádku, tj. $[1;7]$; - dále vybíráme po řadě pole diagonály D_1 v sudých sloupcích zleva doprava; - dále vybíráme po řadě zbylá pole diagonály D_1 v lichých sloupcích zleva doprava; - dále vybíráme zleva doprava všechna pole diagonály D_2; - dále vybíráme zleva doprava všechna pole diagonály D_3, následně D_4 atd.

Pro turnaj o sedmi týmech tak dostaneme následující pořadí zápasů

$$[1;7],\quad [2;3],\quad [4;5],\quad [6;7],\quad [1;2],\quad [3;4],\quad [5;6],$$
$$[1;3],\quad [2;4],\quad [3;5],\quad [4;6],\quad [5;7],\quad [1;4],\quad [2;5],$$
$$[3;6],\quad [4;7],\quad [1;5],\quad [2;6],\quad [3;7],\quad [1;6],\quad [2;7].$$

Úloha 3. Vypište užitím Liborova algoritmu posloupnost zápasů pro turnaj, kterého se účastní 9 týmů a ověřte, že v ní nedojde k výskytu stejného týmu ve dvou po sobě jdoucích zápasech.

Úloha 4. Platí Liborův algoritmus obecně pro libovolný počet přihlášených týmů? Pokud ne, tak pro které? A dokážete sestavit pro tyto případy požadovanou posloupnost sami?

English source

Not available on July 10. If you want to start from English translation, wait until it appears on `https://um.mendelu.cz/math4u/site/` anc copy the English text by hand.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by the European Union