

## Výsledková listina

<b>Termín odevzdání:</b>	<b>30.04.2017 23:59:59</b>
<b>Hodnocení:</b>	<b>6.5000</b>
<b>Max. hodnocení:</b>	<b>5.0000</b> (bez bonusů)
<b>Odevzdaná řešení:</b>	2 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
<b>Nápovědy:</b>	1 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat šablonu třídy, která dokáže sestavovat výsledkové listiny v závislosti na zadaných vstupech - výsledcích zápasů dvojic soupeřů.

Předpokládáme, že máme několik soupeřů. V jednom zápase se utká dvojice soupeřů, výsledek zápasu je nějak popsán (například booleovskou hodnotou, dvojicí počtu bodů, dvojicí časů, ...). Protože chceme mít třídu obecnou, bude výsledek zápasu reprezentován jako generický datový typ. Takto bude do třídy vloženo několik výsledků pro zadané dvojice soupeřů (soupeři jsou jednoznačně identifikováni jménem - řetězcem). Následně chceme zjistit, zda na základě zadaných výsledků lze sestavit jednoznačnou výsledkovou listinu a případně tuto listinu i vypočítat. Trikem je, že při sestavování výsledku chceme vycházet pouze z informací o vítězi/poraženém ze zadaných zápasů.

Výsledková listina je určena jednoznačně, pokud pro každého soupeře  $x_i$  na pozici  $i$  platí:

- $x_i$  porazil všechny soupeře  $x_j$ ,  $i+1 \leq j \leq n$ . Soupeře porazil buď přímo, nebo zprostředkovaně ( $x_i$  porazil  $x_a$ ,  $x_a$  porazil  $x_b$ , ...,  $x_z$  porazil  $x_j$ ,  $i+1 \leq a < b < \dots < j \leq n$ ),
- $x_i$  nevyhrál ani neremizoval se žádným soupeřem  $x_k$ ,  $0 \leq k \leq i-1$ .

Vezměme příklad, kdy bylo zadáno, že A porazí B a B porazí C. Pak je zřejmě správná výsledková listina A B C. Pokud by ale bylo zadáno, že:

A porazí B  
B porazí C  
C porazí A

případně:

A porazí B  
A porazí C

případně:

A porazí B  
B remizuje s C

pak výsledkovou listinu nelze jednoznačně určit (museli bychom dodat informace o dalších zápasech nebo pořadí stanovit i na základě dalších kritérií, to v této úloze nebudeme dělat).

Bylo zmíněno, že zápas je zadaný jako trojice (soupeř1, soupeř2, výsledek), kde výsledek je nějaký datový typ popisující například skóre, časy nebo cokoliv jiného. Výsledek je potřeba převést na informaci o vítězi/remíze/poraženém, to bude mít na starosti porovnávač (komparátor) předaný metodě pro určení výsledkové listiny. Porovnávač je funkce, funktor nebo lambda-výraz, který pro daný výsledek vrací návratovou hodnotu:

- zápornou, pokud soupeř1 prohrál,
- nulovou, pokud soupeř1 a soupeř2 remizovali,
- kladnou, pokud soupeř1 vyhrál.

Celkové rozhraní šablony třídy CContest bude:

šablona je parametrizovaná generickým parametrem `_M`  
tento typ popisuje výsledek zápasu. Pro datový typ `_M` je garantováno, že je přesouvatelný, kopírovatelný a zrušitelný (`CopyConstructible`, `MoveConstructible`, `CopyAssignable`, `MoveAssignable` a `Destructible`). Další vlastnosti typu nejsou garantované, speciálně, není garantované, že typ `_M` má implicitní konstruktor (nemusí být `DefaultConstructible`).  
implicitní konstruktor  
vytvoří prázdnou instanci `CContest`,  
`AddMatch(contestant1, contestant2, result)`

metoda přidá výsledek mezi soupeřem contestant1 a contestant2. Soupeři jsou zadáni v podobě řetězců. Výsledek result je datového typu \_M. Metoda si uloží informaci o tomto zápasu do nějakého úložiště ve své instanci. Volání AddMatch lze řetěžit. Metoda vyvolá výjimku DuplicateMatchException, pokud byl výsledek zápasu mezi contestant1 a contestant2 již dříve zadán.

#### IsOrdered (comparator)

metoda rozhodne, zda vyplněné výsledky zápasů vedou k jednoznačné výsledkové listině nebo ne. Návratovou hodnotou je true pokud je výsledková listina jednoznačná, false pokud ne. Parametrem volání je comparator - funkce, funktor nebo lambda výraz, který bude volán při získávání výsledku zápasu (převede \_M na záporné číslo / nulu / kladné číslo podle popisu výše). Metoda nesmí modifikovat instanci CContest a nesmí házet výjimky.

#### Results (comparator)

metoda vypočte výsledkovou listinu. Výsledkem je STL seznam (std::list) řetězců se jmény soupeřů v pořadí od prvního (celkového vítěze) k poslednímu. Pokud nelze výsledkovou listinu sestavit (není jednoznačná), metoda vyhodí výjimku OrderingDoesNotExistException. Parametr comparator má stejný význam jako u metody IsOrdered.

Odevzdávejte zdrojový kód s implementací šablony třídy CContest. Za základ implementace použijte přiložený zdrojový kód. Pokud v kódu ponecháte bloky podmíněného překladu, lze takový zdrojový kód lokálně testovat a zároveň jej odevzdávat Progtestu.

Hodnocení je rozděleno na povinnou a bonusovou část. V povinné části se testují instance s malým počtem soupeřů a zápasů. Pro jeho úspěšné zvládnutí stačí základní algoritmus pracující v čase počet\_soupeřů x počet\_zápasů. Pro zvládnutí bonusového testu je potřeba použít algoritmus efektivnější.

Pro základní řešení se může hodit vyhledání všech přímo nebo zprostředkovaně poražených soupeřů. Toto dokáže algoritmus BFS (prohledávání do šířky). Pro inspiraci si připomeňte proseminář PA1, ve kterém jsme probírali datové struktury, příklad s cestou v bludišti.

Zdrojový kód s ukázkou práce šablony naleznete v přiloženém archivu.

Vzorová data:

[Download](#)

#### Referenční řešení

**2** **29.04.2017 20:15:58**

[Download](#)

**Stav odevzdání:** Ohodnoceno

**Hodnocení:** 6.5000

##### • Hodnotitel: automat

- Program zkompileován
- Test 'Zakladni test s parametry podle ukazky': Úspěch
  - Dosaženo: 100.00 %, požadováno: 100.00 %
  - Celková doba běhu: 0.001 s (limit: 10.000 s)
  - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi daty': Úspěch
  - Dosaženo: 100.00 %, požadováno: 50.00 %
  - Celková doba běhu: 0.547 s (limit: 9.999 s)
  - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test rychlosti': Úspěch
  - Dosaženo: 100.00 %, požadováno: 100.00 %
  - Celková doba běhu: 4.447 s (limit: 15.000 s)
  - Úspěch v bonusovém testu, hodnocení: 130.00 %
- Celkové hodnocení: 130.00 % (= 1.00 \* 1.00 \* 1.30)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 130.00 %
- Celkem bodů: 1.30 \* 5.00 = 6.50

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	6	--	--	--
	Řádek kódu:	43	7.17 ± 7.40	19	search(int)
	Cykломatická složitost:	9	1.50 ± 2.14	5	AddMatch(const string &,const string &,const _M &)

**1****29.04.2017 20:14:27****Download****Stav odevzdání:** Ohodnoceno**Hodnocení:** 0.0000

- **Hodnotitel: automat**

- ☐ Chyba při základní kompilaci
- Celkové procentní hodnocení: 0.00 %
- Celkem bodů:  $0.00 * 5.00 = 0.00$