



## Problema 11

21 Aprile 2022

### Descrizione

Considera la seguente versione della “conta” che, traendo spunto dal racconto di Giuseppe Flavio, prevede venga servito un commensale ogni tre anziché uno ogni due:

*N sgabelli sono disposti in cerchio attorno a una tavola rotonda, numerati da 1 a N in senso orario (in particolare a sinistra dello sgabello n. N si trova lo sgabello n. 1). Inizialmente, su ciascuno sgabello prende posto un cavaliere con una tazza davanti a sé e il cavaliere seduto al posto n. 1 riceve una brocca di sidro abbastanza capiente per soddisfare tutti. Per servire il sidro si procede applicando le seguenti regole. Identifichiamo con A il cavaliere che in un certo momento ha la brocca; con B quello più prossimo alla sinistra di A; con C quello più prossimo alla sinistra di B. (Detto altrimenti, A, B e C sono i primi tre commensali che si incontrano, nell'ordine, procedendo attorno alla tavola in senso orario a partire da A.) B solleva la tazza di C e la avvicina ad A affinché A possa versargli il sidro. Una volta riempita la propria tazza, C si allontana dalla tavola e va a bersi il sidro sotto una pergola. Quindi A passa la brocca al commensale che si trovava alla sinistra di C prima che quest'ultimo si allontanasse. Di volta in volta chi riceve la brocca e i due commensali successivi si comportano allo stesso modo, finché le regole non possono più essere applicate perché solo gli ultimi due cavalieri rimangono seduti a tavola.*

*Poiché la brocca contiene sidro in abbondanza, i due cavalieri che dovranno servirsi per ultimi avranno l'opportunità di vuotarla riempiendo ripetutamente le proprie tazze. Immagina che tu e un/a amico/a siete due dei cavalieri, quali sgabelli scegliereste, identificati dai due corrispondenti numeri, per avere questo privilegio?*

Procedendo in modo analogo a quanto visto a lezione, ridefinisci opportunamente la classe `RoundTable` e scrivi un programma che risolve il problema proposto. Il nuovo protocollo pubblico della classe è così specificato:

<code>rt = new RoundTable(n)</code>	: <code>RoundTable</code>	costruttore della disposizione iniziale con $(int) n \geq 2$ cavalieri
<code>rt.numberOfKnights()</code>	: <code>int</code>	numero di cavalieri a tavola
<code>rt.servingKnights()</code>	: <code>IntSList</code>	coppia (lista di due elementi) di cavalieri che servono il terzo se c'è
<code>rt.serveNeighbour()</code>	: <code>RoundTable</code>	disposizione risultante dopo aver servito il prossimo cavaliere, che esce
<code>rt.passJug()</code>	: <code>RoundTable</code>	disposizione risultante dopo aver passato la brocca

In particolare, realizza una versione “efficiente” delle procedure del protocollo utilizzando due liste di interi (`IntSList`) come nell'esempio più sofisticato discusso a lezione.

**Esempi** (assumendo che la procedura principale sia denominata `josephus` come nel codice disponibile online)

<code>josephus(2)</code>	→	<code>(1, 2)</code>
<code>josephus(3)</code>	→	<code>(1, 2)</code>
<code>josephus(4)</code>	→	<code>(4, 1)</code>
<code>josephus(5)</code>	→	<code>(2, 4)</code>
<code>josephus(6)</code>	→	<code>(5, 1)</code>
<code>josephus(7)</code>	→	<code>(1, 4)</code>
<code>josephus(8)</code>	→	<code>(4, 7)</code>
<code>josephus(12)</code>	→	<code>(5, 10)</code>
<code>josephus(1500)</code>	→	<code>(338, 905)</code>