

Rešitev četrte domače naloge (Na šejkovem dvoru)

Začetek ni posebej težak. S pomočjo objekta tipa `Scanner` preberemo število oznak, ciljno vsoto in posamezne oznake moških in žensk:

```
import java.util.Scanner;

public class SejkNaivno {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int stOznak = sc.nextInt();
        int ciljnaVsota = sc.nextInt();

        int[] oznake = new int[stOznak];
        for (int i = 0; i < stOznak; i++) {
            oznake[i] = sc.nextInt();
        }
        ...
    }
}
```

Sedaj moramo prešteti pare oznak, katerih vsota je enaka ciljni vsoti. To lahko dosežemo tako, da se z dvojno zanko sprehodimo po vseh parih oznak in za vsak par preverimo njegovo vsoto. Če je ta enaka ciljni vsoti, povečamo števec parov, ki ga moramo na začetku seveda postaviti na ničlo. Na koncu števec izpišemo.

```
long stParov = 0;
for (int i = 0; i < stOznak; i++) {
    for (int j = 0; j < stOznak; j++) {
        if (oznake[i] + oznake[j] == ciljnaVsota) {
            stParov++;
        }
    }
}
System.out.println(stParov);
```

Takšna rešitev je sicer pravilna, žal pa ni dovolj učinkovita. Če imamo na vходу 10^6 oznak, moramo preveriti kar 10^{12} parov, to pa je več, kot današnji računalniki zmorejo v eni sekundi. V besedilu naloge je navedeno, da vse oznake pripadajo intervalu $[0, 1000]$. To pomeni, da bodo pri velikem številu udeležencev — torej v primerih, ki nam dejansko povzročajo sive lase — številni moški in ženske imeli iste oznake, zato bo v tabeli **oznake** veliko elementov imelo enake vrednosti. K sreči pa je množico oznak udeležencev mogoče predstaviti varčneje. Definiramo tabelo **frekvenca**, v kateri element i podaja število moških oz. žensk z oznako i . Dolžina tabele mora znašati najmanj 1001 (tako bo element z indeksom 1000 vseboval število moških oz. žensk z največjo možno oznako), kot bo postalo jasno kasneje, pa je udobneje vzeti dolžino 2000 — toliko, kot znaša največja možna ciljna vsota. Tabelo **frekvenca** napolnimo tako, da se sprehodimo po oznakah in po branju oznake **oznaka** povečamo element **frekvenca[oznaka]**.

```
import java.util.Scanner;

public class Sejk {
```

```

private static final int VELIKOST_TABELE = 2000;

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int stOznak = sc.nextInt();
    int ciljnaVsota = sc.nextInt();

    int[] frekvenca = new int[VELIKOST_TABELE];
    for (int i = 0; i < stOznak; i++) {
        int oznaka = sc.nextInt();
        frekvenca[oznaka]++;
    }
    ...
}
}

```

Kako lahko sedaj določimo število parov s podano ciljno vsoto v ? Moške z oznako 1 moramo »popariti« z ženskami z oznako $v - 1$. Ker imamo `frekvenca[1]` moških z oznako 1 in `frekvenca[v - 1]` žensk z oznako $v - 1$, je število tovrstnih parov enako

`frekvenca[1] * frekvenca[v - 1]`

Moške z oznako 2 moramo popariti z ženskami z oznako $v - 2$. Teh parov je

`frekvenca[2] * frekvenca[v - 2]`

In tako naprej. Število vseh parov torej izračunamo kot vsoto

`frekvenca[1] * frekvenca[v - 1] +`
`frekvenca[2] * frekvenca[v - 2] +`
`... +`
`frekvenca[v - 1] * frekvenca[1]`

ki se v java prepíše takole:

```

long stParov = 0;
for (int oznaka = 1; oznaka < ciljnaVsota; oznaka++) {
    stParov += ((long) frekvenca[oznaka]) *
               ((long) frekvenca[ciljnaVsota - oznaka]);
}

```

Ker lahko zmnožek frekvenc preseže zgornjo mejo za tip `int`, moramo vsaj en faktor pretvoriti v tip `long`, sicer bi java števili zmnožila v okviru meja za tip `int`. S pretvorbo tipa pa izsilimo množenje v okviru meja za tip `long`.

Oglejmo si celotno kodo izboljšane rešitve:

```

import java.util.Scanner;

public class Sejk {

    private static final int VELIKOST_TABELE = 2000;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int stOznak = sc.nextInt();
        int ciljnaVsota = sc.nextInt();
    }
}

```

```

    int[] frekvenca = new int[VELIKOST_TABELE];
    for (int i = 0; i < stOznak; i++) {
        int oznaka = sc.nextInt();
        frekvenca[oznaka]++;
    }

    long stParov = 0;
    for (int oznaka = 1; oznaka < ciljnaVsota; oznaka++) {
        stParov += ((long) frekvenca[oznaka] *
                    ((long) frekvenca[ciljnaVsota - oznaka]));
    }
    System.out.println(stParov);
}
}

```