

Meno: Michal Korbela  
trieda: sexta  
škola: Gymnázium J.J. Bánovce  
úloha č.4

najskôr si musíme uvedomiť, ako dosiahneme najmenší počet balíkov, tak aby v každom boli rôzne čísla. Zoberme si čísla, ktorých je najviac. Tolko vytvoríme balíkov, lebo keby ich bolo menej, tak podľa Dirichleta v aspoň jednom musia byť dve rovnaké. Vieme, že už čísel, ktorých je väčší počet ako týchto nieje, keďže sme vybrali ten najväčší, tak tento počet bude určovať počet balíkov. Keby sme ich vytvorili viac, tak je nám to len na škodu. A ako optimalizovať balíky, aby bolo v každom čo najmenej?? tak určite môžeme rozdeliť čísla, že si pripravíme balíky a začneme do nich ukladať postupne rovnaké čísla. Najskôr do prvého dáme 1, potom do druhého dáme tiež jedno, atď. a tiež do posledného dáme 1. Takto budeme mať v každom balíku po jedno číslo. Vždy keď nám dojdú rovnaké, tak začneme ukladať ďalšie, iné, ale zase rovnaké, a keď nám aj tie dojdú, tak zase rovnaké. Takto poukladáme všetky čísla, a zistíme, že vo všetkých balíkoch je  $n/p$ , kde  $n$  je počet čísel a  $p$  je počet balíkov zaokrúhlené nadol, a v niektorých je  $n/p$  zaokrúhlené nadol + 1. Takýto balík však nemusí byť ani jeden.

```
#include <iostream>
#include <string>
#include <stdlib.h>
#include <sstream>
#include <math.h>
#include <stdio.h>
using namespace std;

int compare (const void * a, const void * b)
{
    return ( *(int*)a - *(int*)b );           //funkcia potrebná na
quicksort
}

int main(){
    long long n;
    cin>>n;

    long long * pole=new long long[n+1];
    for(long i=0;i<n;i++)
        cin>>pole[i];                        //načítanie

    qsort (pole, n, sizeof(long long), compare);    //utriedíme
    prvky od najmenšieho po najväčší, a vždy rovnaké čísla sú pri sebe, takže nieje
    problém zistiť, koľko je ktorých

    long long pos=0,naj=0;
    long long pocet=1;
```

```

while(pos<n){
    if(pole[pos]==pole[pos+1])pocet++;           //pokiaľ sú
    stále čísla rovnaké pokračujeme - prehliadame pole
    else{
        if(pocet>naj)naj=pocet;                 // ak je čísel viac
        ako tých predošlých, tak si hodnotu zapíšeme
        pocet=1;                                //a vynulujeme pole
        }
        pos++;
    }

    int zvys=0;
    if (n%naj!=0) zvys=1;                        //ak sa všetky čísla
    nedajú rozdeliť rovnako do všetkých balíkov

        cout<<naj<<"\n"<<(n/naj)+ zvys<<endl;    //tak
    pripočítame jedna k výsledku, ak nie, tak nepripočítame

        return 0;
    }

```

Čo sa týka časovej a Pamäťovej zložitosti, tak časová je  $O(n \cdot \log N)$  a to preto, lebo použijeme quicksort, ktorý má časovú zložitosť  $O(N \cdot \log N)$  a väčšia sa už nikde nenachádza a pamäťová - lineárna, pretože si musíme zapísať všetky čísla - priamo v načítavaní by sa dal určiť tiež počet čísel, ale to by sme si museli zapísať ich počty, čo by v prípade kde by boli čísla všetky rozne bola tiež lineárna, takže menej sa to nedá. Takže  $O(N) - n$  je počet čísel.