

Meno: Michal Korbela  
trieda: sexta  
škola: Gymnázium J.J. Bánovce  
úloha č.5

Najskôr som rozmýšľal použiť metódu brute-force – rekurzívne všetky možnosti prehľadať.  
Avšak n(počet čísel) 7 mi to vyhodilo za 2 sekundy  
pre n=8 za 2 minúty a pre n=9 za 20 min a povedal som si, že keď to pôjde takto ďalej, tak to bude robiť aspoň 2 roky, a to nestihnem do deadlinu.

Preto som vymyslel osekáný brute-force.

Nájde všetky možnosti pre n, a pre n+1 zoberiem ako základ z nich len x možností, ktoré dávajú čo najväčší základ.

```
#include <iostream>
#include <queue>
#include <vector>
#include <time.h>
#include <stdio.h>
```

```
using namespace std;
vector<priority_queue<vector<int>,vector<vector<int> >,less<vector<int> > > > F; //fronta,
ktorá vracia najväčší vektor
int pocet=20000; //počet koľko naj vektorov má odskúšať
```

```
int main(){
F.resize(32); //viac asi potrebovať nebudeme
vector<int> h; //vytvoríme základ
h.push_back(3); //prvý prvok tvorí vždy číslo, kam sa vieme dostať
h.push_back(1); //2. a ďalšie čísla – začíname s 1
F[1].push(h);
```

```
int n;
cin>>n; //po koľko chceme vypísať čísla +1 čiže dávame 29
time_t rawtime;
struct tm * timeinfo;

time ( &rawtime );
timeinfo = localtime ( &rawtime );
cout<<asctime (timeinfo)<<endl; //vypíšeme začiatok

for(int i=1; i<=n; i++){
```

```

int j=0;
while(j<pocet && F[i].size()>0){           //pokiaľ nieje fronta prázdna,alebo sme vyskúšali ešte
menej ako pocet prvkov
vector<int>::iterator it,suce,it1,pos,sumx,konx,kon,c,a1,a2,a3;
vector<int>::iterator u;
vector<int> x,y;
x=F[i].top();                             //zoberieme najväčší vektor

y=x;

y.push_back(0);

sumx=x.begin();                           //nastavíme iteratory
konx=x.end()-1;
suce=y.begin();
kon=y.end()-1;

int zac=*(kon-1),koni=(*suce);

for(int k=zac+1; k<=koni+1; k++){          //postupne skúšame všetky možné čísla – môžeme však
len od najväčšieho čísla – menšie nemá význam, až po maximálnu hodnotu, kam sa vieme dostať
(*suce)=(*sumx);
(*kon)=k;

int ok=1;
do{
ok=0;                                     //vypočítame maximum, kam sa vieme dostať
for(a1=y.begin()+1; a1<y.end(); a1++){
if((*a1)==(*suce)+1) {ok=1; (*suce)++;}

for(a2=y.begin()+1; a2<y.end(); a2++){
if((*a1+*a2)==(*suce)+1) {ok=1; (*suce)++;}

for(a3=y.begin()+1; a3<y.end(); a3++){
if((*a1+*a2+*a3)==(*suce)+1) {ok=1; (*suce)++;}
}
}
}

}while (ok==1);

```

```

F[i+1].push(y);                //pridáme náš výsledok do fronty
}
F[i].pop();                    //a odstránime, ten ktorý sme skúšali
j++;
}
vector<int> sa;
sa=F[i+1].top();               //vypíšeme najväčší vektor – tam kam sa vieme najviac dostať
vector<int>::iterator s;
for(s=sa.begin(); s<sa.end(); s++)
cout<<(*s)<<" ";
cout<<endl;
}
time_t rawtime1;              //vypíšeme konečný čas
struct tm * timeinfo1;
time ( &rawtime1 );
timeinfo1 = localtime ( &rawtime1 );
cout<<asctime (timeinfo1)<<endl;

}

```

má to veľké zložitosti – čas  $O(3^n)$  pre každé  $n$  od 1 až po 30.

pamäť to má tiež veľkú  $O(3^n)$  – aj keď v skutočnosti je oveľa menšia, pretože berieme len 20000 prvkov

pretože ak zoberieme max číslo, ktoré tam môžeme dať na koniec, tak to bude max  $3 \cdot$  posledné číslo, takže pre  $n$ -té číslo to bude  $3^n$

Pre pocet=10000 to bežalo 4 hod  
potom som to pustil pre pocet 20000, ale keď som k tomu ráno prišiel, tak som našiel proces zabitý, tak som z toho zobral aspoň po  $n=25$  – iba to to dalo.

A dostal som skóre viac ako 758.