Uzorak briljantnosti

(slučajni podskupovi i permutacije)

Ivan Čeh

Prirodoslovno- matematički fakultet

Seminar iz kolegija „Oblikovanje i analiza algoritama”

Zagreb, 6. siječnja 2017.

Motivacija

- računalne kartaške igre
- Karte se ne smiju ponavljati!

Podskupovi i permutacije

- Neka je S konačni skup, $|S| = n,\ k \in \mathbb{N},\ k \leq n$
- k-podskup ili k-kombinacija u S je k-člani podskup skupa S
- Ukupni broj k-podskupova nekog n-članog skupa je $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$

Podskupovi i permutacije

- k-permutacija (bez ponavljanja) skupa S je uređena k-torka različitih brojeva iz skupa *S.*
- Ukupan broj k-permutacija nekog n-članog skupa je $n^{\underline{k}} = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot (n-k+1)$

Slučajni podskupovi i permutacije

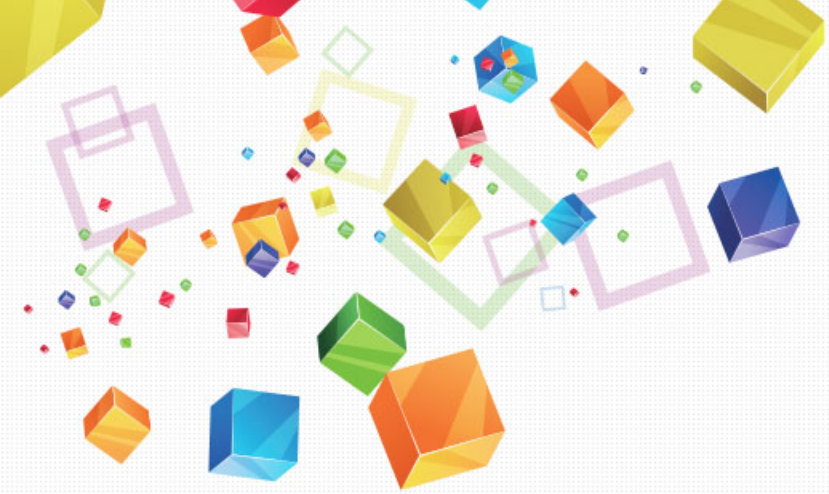
- Problem:
  - Generirati slučajni k-podskup skupa {1, 2, 3, ⋯, n}
    - Poredak elemenata nije bitan (u našem slučaju uzlazno sortiran).
  - Generirati slučajnu k-permutaciju skupa {1, 2, 3, ⋯, n}
    - Poredak elemenata mora također biti slučajan.

Zahtjevi

- Vjerojatnost za svaka dva moguća podskupa/ permutacije mora biti jednaka!
  - Svaki k-podskup se pojavljuje s vjerojatnošću $\frac{1}{\binom{n}{k}} = \frac{k! \cdot (n-k)!}{n!}$
  - Svaka k-permutacija se pojavljuje s vjerojatnošću $\frac{1}{n^{\underline{k}}}$
- Što manja vremenska i memorijska složenost

Slučajni brojevi

- Funkcija rand() iz biblioteke <stdlib.h>
  - Vraća slučajni broj iz intervala $[0, RAND\_MAX\rangle$ (RAND_MAX je najčešće $2^{15} = 32768$)
- Biblioteka <random> iz C++11 standarda
  - Mersenne Twister algoritam
  - mogućnost određivanja granica



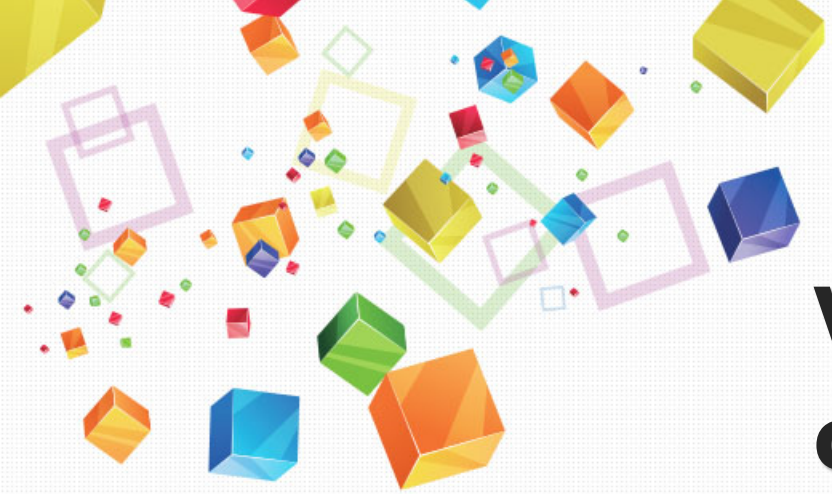
Slučajni brojevi

```
4   #include <random>
5
6   class Random {
7   public:
8       Random() = default;
9       Random(std::mt19937::result_type seed) : eng(seed) {}
10      int operator() (int min, int max)
11      {
12          return std::uniform_int_distribution<int>{min, max}(eng);
13      }
14
15  private:
16      std::mt19937 eng{std::random_device{}()};
17  };
```

Vremenska zahtjevnost operacija

```
10      Random random;
11      int i,s1=0,s2=0,s3=0;
12      SYSTEMTIME t0,t1,t2,t3,t4;
13      std::set<int> S;
14      GetSystemTime(&t0);
15      for(i=0;i<10000000;++i)
16          s1+=i;
17      GetSystemTime(&t1);
18      for(i=0;i<10000000;++i)
19          s2+=rand();
20      GetSystemTime(&t2);
21      for(i=0;i<10000000;++i)
22          s3+=random(0,i);
23      GetSystemTime(&t3);
24      for(i=0;i<10000000;++i)
25          S.insert(i);
26      GetSystemTime(&t4);
27      std::cout<<msrazlika(t1,t0)<<' '<<msrazlika(t2,t1)
28          <<' '<<msrazlika(t3,t2)<<' '<<msrazlika(t4,t3);
```

```
39 176 954 25213
Process returned 0 (0x0)   execution time : 34.195 s
Press any key to continue.
```



Vremenska zahtjevnost operacija

- Određivanje slučajnog broja je znatno vremenski zahtjevnije od jednostavnih aritmetičkih operacija.
- Operacije na STL-ovim spremnicima podataka mogu biti znatno zahtjevnije i od određivanja slučajnog broja.
- Kako izbjeći ponavljanja brojeva pomoću funkcija za generiranje slučajnih brojeva?

Algoritam pokušaja

```
S = prazan skup
dok (velicina (S) < k)
        t = random [1, n]
        ako t ∉ S
                umetni t u S
vrati S
```

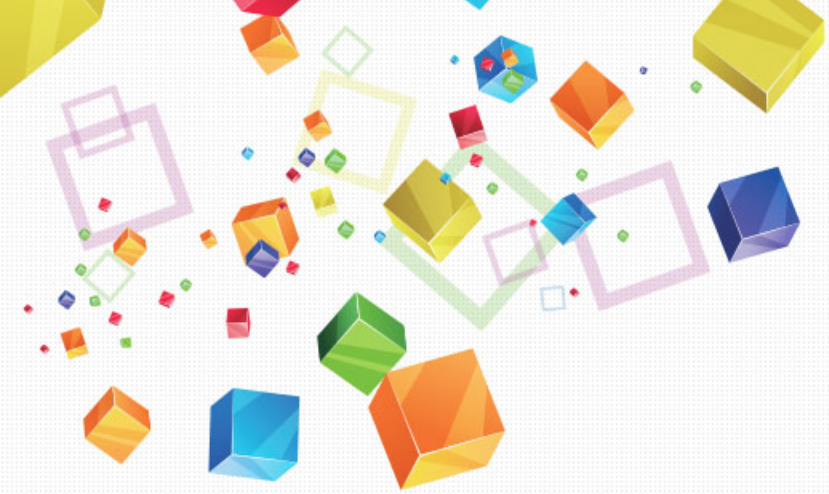
Algoritam pokušaja

prednosti

- lako shvatljiv i jednostavan za implementaciju

mane

- moguć je velik broj ponovljenih poziva funkcije random
- najgora vremenska složenost je $+\infty$



Algoritam pokušaja za generiranje slučajnog podskupa

```
5   std::set<int> slucajni_podskup_pokusajima (int n, int k, Random& random)
6   {
7       std::set<int> S;
8       int i, t;
9       for (i=0; i<k; ++i)
10          do
11              t = random(0, n-1);
12          while (!S.insert(t).second); // dok je element t vec otprije bio u S
13      return S;
14  }
```

Floydov algoritam za generiranje slučajnog podskupa, rekurzivno

```
podskup (n, k)
        ako je k=0
                vrati prazan skup
        inače
                S = podskup(n-1, k-1)
                t = random [1, n]
                ako je t ∈ S
                        umetni n u S
                inače
                        umetni t u S

vrati S
```

Floydov algoritam za generiranje slučajnog podskupa, iterativno

```
S = prazan skup
za i od n-k+1 do n
        t = random [1, i]
        ako je t ∈ S
                umetni i u S
        inače
                umetni t u S
vrati S
```

Floydov algoritam- distribucija
(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- baza indukcije- Očito vrijedi za n = 1
- korak- Pretpostavimo da tvrdnja vrijedi za n-1 i sve k' ∈ {0, 1, 2, ⋯, n-1}
  Neka je S neki n-člani skup
  - Ako je n ∈ K onda je $P(K) = \frac{(k-1)! \cdot (n-k)!}{(n-1)!} \cdot \frac{k}{n} = \frac{k! \cdot (n-k)!}{n!}$
  - Ako n ∉ K onda je $P(K) = k \cdot \frac{(k-1)! \cdot (n-k)!}{(n-1)!} \cdot \frac{1}{n} = \frac{k! \cdot (n-k)!}{n!}$

Floydov algoritam za generiranje slučajnog podskupa

```
 7  void floyd_rekurzija (std::set<int> &S, int n, int k, Random& random)
 8  {
 9      int t;
10      if(k == 0)
11          return; // prazan skup
12      else
13      {
14          floyd_rekurzija (S, n-1, k-1, random);
15          t = random(0, n-1);
16          if (S.insert(t).second == false) // ako je t vec otprije bio u S
17              S.insert(n-1);
18      }
19  }
20
21  std::set<int> slucajni_podskup_floyd_rek (int n, int k, Random& random)
22  {
23      std::set<int> S;
24      floyd_rekurzija (S, n, k, random);
25      return S;
26  }
27
28  std::set<int> slucajni_podskup_floyd_nerek (int n, int k, Random& random)
29  {
30      std::set<int> S;
31      int i,t;
32      for (i=n-k; i<n; ++i)
33      {
34          t = random(0, i);
35          if (S.insert(t).second == false) // ako je t vec otprije bio u S
36              S.insert(i);
37      }
38      return S;
39  }
```

Floydov algoritam za generiranje slučajnog podskupa, poljem bitova

```
41  std::vector<bool> slucajni_podskup_floyd_bitset (int n, int k, Random& random)
42  {
43      std::vector<bool> V;
44      int i,t;
45      V.resize(n,false);
46      for (i=n-k; i<n; ++i)
47      {
48          t = random(0, i);
49          if (V[t]) // ako je t vec otprije bio u S
50              V[i]=true;
51          else
52              V[t]=true;
53      }
54      return V;
55  }
```

Algoritam pokušaja za permutacije

X je k-člani niz u koji upisujemo permutaciju
za i od 1 do k
    ponavljaj
        t = random [1, n]
    dok ne nađemo t koji nije u X
    X [i] = t
vrati S

Algoritam pokušaja za permutacije

- U slučaju n-permutacije n-članog skupa očekivani broj poziva funkcije random je $\sum_{i=1}^{n} \frac{n}{i} \in \theta(n \log n)$
- U najgorem slučaju algoritam nikad ne staje.

Algoritam pokušaja za permutacije

```
16  std::vector<int> slucajna_permutacija_pokusajima (int n, int k, Random& random)
17  {
18      std::vector<int> V;
19      std::set<int> S;
20      V.resize(k);
21      int i, t;
22      for (i=0; i<k; ++i)
23      {
24          do
25              t = random(0, n-1);
26          while (!S.insert(t).second); // dok je element t vec otprije bio u S
27          V[i] = t;
28      }
29      return V;
30  }
```

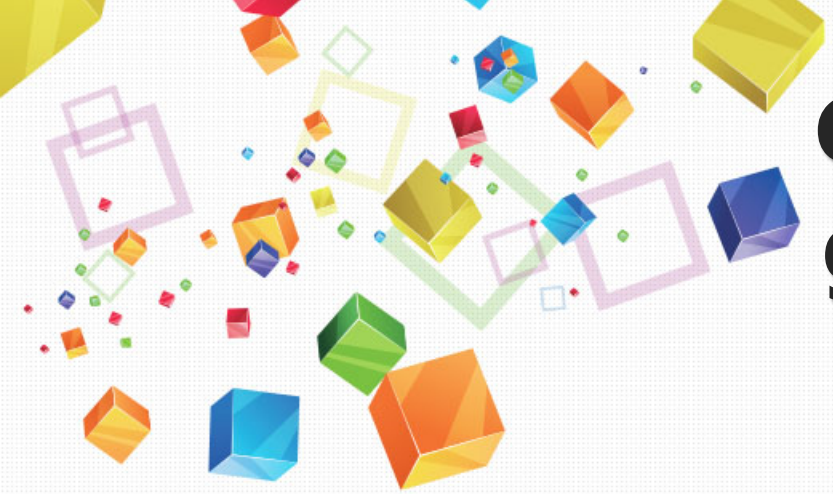
Generiranje permutacija generiranjem podskupa i miješanjem

u niz X upisujemo k-podskup od S (koji u pravilu nije slučajno poredan)

```
za i od 2 do k
        t = random [1, i]
        zamijeni (X[i], X[t])
vrati X
```

Generiranje permutacija generiranjem podskupa i miješanjem

```
49  std::vector<int> slucajna_permutacija_floyd_podskup_i_mijesanje (int n, int k, Random& random)
50  {
51      std::set<int> S;
52      std::vector<int> V;
53      int i=0,t;
54      S = slucajni_podskup_floyd_nerek (n, k, random);
55      V.resize(k);
56      for(int x:S) // Kopiranje S u V u sortiranom poretku.
57          V[i++]=x;
58      for (i=1; i<k; ++i)
59      {
60          t = random(0, i);
61          zamijeni (V[i], V[t]);
62      }
63      return V;
64  }
```

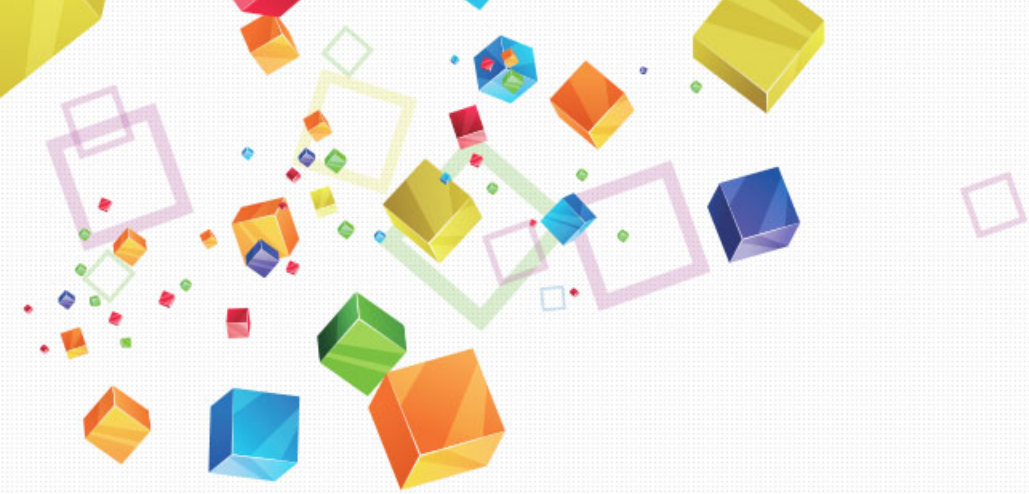


Generiranje permutacija generiranjem podskupa i miješanjem

- za k-člani podskup n-članog skupa funkcija random se poziva 2k puta (k za generiranje podskupa, k za miješanje)

Generiranje permutacija miješanjem

```
za i od 1 do n
        X[i] = i;
za i od 1 do k
        t = random [i, n];
        zamijeni (X[i], X[t])
vrati X
```



Generiranje permutacija miješanjem

```
66  std::vector<int> slucajna_permutacija_mijesanjem (int n, int k, Random& random)
67  {
68      std::vector<int> V;
69      int i, t;
70      V.resize(n);
71      for (i=0; i<n; ++i)
72          V[i] = i;
73      for (i=0; i<k; ++i)
74      {
75          t = random(i, n-1);
76          zamijeni(V[i], V[t]);
77      }
78      V.resize(k);
79      return V;
80  }
```

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

  2 1 9 6 3

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

  2 1 9 6 3

1 2 3 4 5 6 7 8 9 i = 1, t = 2

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

  2 1 9 6 3

2 1 3 4 5 6 7 8 9 i = 2, t = 2

1 2 3 4 5 6 7 8 9 i = 1, t = 2

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

  2 1 9 6 3

2 1 3 4 5 6 7 8 9	i = 3, t = 9
2 1 3 4 5 6 7 8 9	i = 2, t = 2
1 2 3 4 5 6 7 8 9	i = 1, t = 2

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

2 1 9 4 5 6 7 8 3	i = 4, t = 6
2 1 3 4 5 6 7 8 9	i = 3, t = 9
2 1 3 4 5 6 7 8 9	i = 2, t = 2
1 2 3 4 5 6 7 8 9	i = 1, t = 2

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

2 1 9 6 5 4 7 8 3	i = 5, t = 9
2 1 9 4 5 6 7 8 3	i = 4, t = 6
2 1 3 4 5 6 7 8 9	i = 3, t = 9
2 1 3 4 5 6 7 8 9	i = 2, t = 2
1 2 3 4 5 6 7 8 9	i = 1, t = 2

Generiranje permutacija miješanjem – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3	
2 1 9 6 3 4 7 8 5	
2 1 9 6 5 4 7 8 3	i = 5, t = 9
2 1 9 4 5 6 7 8 3	i = 4, t = 6
2 1 3 4 5 6 7 8 9	i = 3, t = 9
2 1 3 4 5 6 7 8 9	i = 2, t = 2
1 2 3 4 5 6 7 8 9	i = 1, t = 2

Generiranje permutacija miješanjem

prednosti

- samo k poziva funkcije random
- jedini spremnik podataka koji se koristi je vektor
- nema dodatnih provjera je li element već sadržan u permutaciji

mane

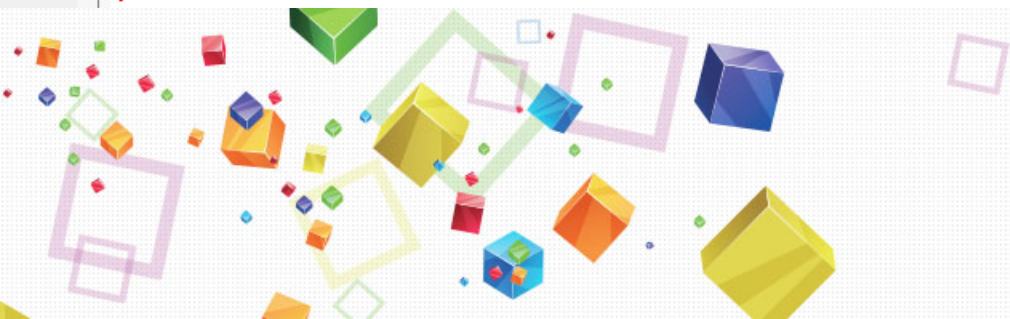
- spor u slučaju k << n (inicijalizacija niza zahtijeva n koraka)

Floydov algoritam za generiranje slučajne permutacije

```
L = prazna lista
za i od n-k+1 do n
        t = random [1, i]
        ako t ∈ L
                umetni i u L nakon t
        inace
                stavi t na pocetak L
vrati L
```

Floydov algoritam za generiranje slučajne permutacije

```
82  std::forward_list<int> slucajna_permutacija_floyd (int n, int k, Random& random)
83  {
84      std::map<int, std::forward_list<int>::iterator> M;
85      std::forward_list<int> L;
86      int i,t;
87      std::map<int, std::forward_list<int>::iterator>::iterator it;
88      for(i=n-k; i<n; ++i)
89      {
90          t = random(0, i);
91          it = M.find(t);
92          if (it == M.end()) // ako t nije bio u M
93          {
94              L.push_front(t);
95              M[t] = L.begin();
96          }
97          else
98              M[i] = L.insert_after(it->second, i);
99      }
00      return L;
01  }
```



Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

prazna lista

i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

3

prazna lista

i = 6, t = 6

i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

6 3	i = 7, t = 1
3	i = 6, t = 6
prazna lista	i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

1 6 3	i = 8, t = 2
6 3	i = 7, t = 1
3	i = 6, t = 6
prazna lista	i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3

2 1 6 3	i = 9, t = 1
1 6 3	i = 8, t = 2
6 3	i = 7, t = 1
3	i = 6, t = 6
prazna lista	i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije – distribucija

(dokaz da je zahtjev o vjerojatnostima zadovoljen)

- Svaka permutacija se može dobiti na točno jedan način
- Primjer (n=9, k=5)

2 1 9 6 3	
2 1 6 3	i = 9, t = 1
1 6 3	i = 8, t = 2
6 3	i = 7, t = 1
3	i = 6, t = 6
prazna lista	i = 5, t = 3

Floydov algoritam za generiranje slučajne permutacije

prednosti

- samo k poziva funkcije random
- brz i u slučaju k << n

mane

- koriste se složene strukture podataka, operacije na njima su vremenski zahtjevne

Usporedba algoritama (za navedenu implementaciju)

	Algoritam	Najgori broj poziva funkcije random	Prosječan broj poziva funkcije random	Najgora složenost	Prosječna složenost
Podskupovi	Algoritam pokušaja	$+\infty$	$\sum_{i=0}^{k-1} \frac{n}{(n-i)}$	$+\infty$	$\theta\left(\log k \sum_{i=0}^{k-1} \frac{n}{(n-i)}\right)$
	Floydov algoritam	k	k	$\theta(k \log k)$ ili $\theta(n+k)$	$\theta(k \log k)$ ili $\theta(n+k)$
Permutacije	Algoritam pokušaja	$+\infty$	$\sum_{i=0}^{k-1} \frac{n}{(n-i)}$	$+\infty$	$\theta\left(\log k \sum_{i=0}^{k-1} \frac{n}{(n-i)}\right)$
	Generiranje podskupa (Floyd) i miješanje	$2k$	$2k$	$\theta(k \log k)$	$\theta(k \log k)$
	Algoritam miješanja	k	k	$\theta(n+k)$	$\theta(n+k)$
	Floydov algoritam	k	k	$\theta(k \log k)$	$\theta(k \log k)$

Test distribucija (primjer koda)

```
20  void provjeri_distribuciju()
21  {
22      Random random(time(nullptr));
23
24      std::map < std::set<int> , int> M;
25      int i, n, k;
26      std::map < std::set<int> , int>:: iterator it;
27      std::cin >> n >> k;
28      for (i=0; i<100000; ++i)
29          ++M[slucajni_podskup_floyd_nerek(n,k,random)];
30      for (it=M.begin(); it!=M.end(); ++it)
31      {
32          ispisi(it->first);
33          std::cout << ' ' << it->second << std::endl;
34      }
35  }
```

Test distribucija, rezultati

```
6 3
0 1 2   4889
0 1 3   4986
0 1 4   5051
0 1 5   5044
0 2 3   4920
0 2 4   4925
0 2 5   5080
0 3 4   4947
0 3 5   5041
0 4 5   5001
1 2 3   4951
1 2 4   5028
1 2 5   5039
1 3 4   5123
1 3 5   5004
1 4 5   4911
2 3 4   4959
2 3 5   5004
2 4 5   5099
3 4 5   4998
```

```
6 3
0 1 2   4988
0 1 3   4981
0 1 4   5048
0 1 5   4931
0 2 3   5011
0 2 4   5084
0 2 5   5012
0 3 4   5035
0 3 5   4963
0 4 5   4961
1 2 3   5112
1 2 4   5059
1 2 5   4915
1 3 4   4980
1 3 5   4865
1 4 5   5035
2 3 4   5078
2 3 5   4992
2 4 5   4930
3 4 5   5020
```

```
6 3
0 1 2   5059
0 1 3   5012
0 1 4   4925
0 1 5   5074
0 2 3   4930
0 2 4   5013
0 2 5   5011
0 3 4   5017
0 3 5   4971
0 4 5   5007
1 2 3   5015
1 2 4   5096
1 2 5   4944
1 3 4   5037
1 3 5   5018
1 4 5   4934
2 3 4   4907
2 3 5   5009
2 4 5   4946
3 4 5   5075
```

```
6 3
3 4 5   4998
2 4 5   5025
2 3 5   5094
2 3 4   4811
1 4 5   4901
1 3 5   5061
1 3 4   5126
1 2 5   5187
1 2 4   5026
1 2 3   4981
0 4 5   4978
0 3 5   4897
0 3 4   5016
0 2 5   4937
0 2 4   4880
0 2 3   4964
0 1 5   5084
0 1 4   5056
0 1 3   4969
0 1 2   5009
```

```
4 3
0 1 2   4229
0 1 3   4151
0 2 1   4231
0 2 3   4216
0 3 1   4105
0 3 2   4155
1 0 2   4139
1 0 3   4148
1 2 0   4128
1 2 3   4001
1 3 0   4168
1 3 2   4127
2 0 1   4147
2 0 3   4148
2 1 0   4235
2 1 3   4262
2 3 0   4071
2 3 1   4230
3 0 1   4027
3 0 2   4143
3 1 0   4222
3 1 2   4188
3 2 0   4257
3 2 1   4272
```

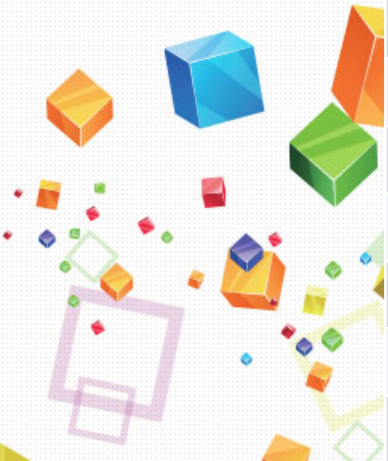
```
4 3
0 1 2   4195
0 1 3   4205
0 2 1   4275
0 2 3   4158
0 3 1   4076
0 3 2   4231
1 0 2   4122
1 0 3   4171
1 2 0   4126
1 2 3   4106
1 3 0   4149
1 3 2   4107
2 0 1   4204
2 0 3   4159
2 1 0   4186
2 1 3   4185
2 3 0   4169
2 3 1   4191
3 0 1   4128
3 0 2   4203
3 1 0   4272
3 1 2   4094
3 2 0   4136
3 2 1   4152
```

```
4 3
0 1 2   4076
0 1 3   4138
0 2 1   4086
0 2 3   4117
0 3 1   4229
0 3 2   4126
1 0 2   4081
1 0 3   4159
1 2 0   4251
1 2 3   4092
1 3 0   4287
1 3 2   4236
2 0 1   4313
2 0 3   4110
2 1 0   4196
2 1 3   4239
2 3 0   4034
2 3 1   4096
3 0 1   4138
3 0 2   4254
3 1 0   4150
3 1 2   4167
3 2 0   4232
3 2 1   4193
```

```
4 3
0 1 2   4217
0 1 3   4177
0 2 1   4112
0 2 3   4191
0 3 1   4230
0 3 2   4168
1 0 2   4156
1 0 3   4058
1 2 0   4254
1 2 3   4360
1 3 0   4120
1 3 2   4106
2 0 1   4134
2 0 3   4057
2 1 0   4176
2 1 3   4260
2 3 0   4196
2 3 1   4129
3 0 1   4167
3 0 2   4116
3 1 0   4234
3 1 2   4181
3 2 0   4102
3 2 1   4099
```

Vremenska složenost funkcija (ms)

		permutacije			
n	k	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
10	10	0,03099	0,02779	0,00372	0,04261
20	20	0,06842	0,05438	0,00528	0,08585
50	50	0,2118	0,14174	0,00918	0,22332
100	100	0,4631	0,285	0,0156	0,4544
200	200	1,0623	0,6035	0,0292	0,9234
500	500	3,1354	1,5339	0,0687	2,3683
1.000	1.000	7,081	3,261	0,14	4,907
2.000	2.000	16,088	6,663	0,27	10,083
5.000	5.000	47,08	17,68	0,67	26,17
10.000	10.000	104,65	36,36	1,31	53,33
20.000	20.000	243,92	76,68	2,65	111,16
50.000	50.000	820,5	207,2	6,9	294
100.000	100.000	2017,7	436,3	13,1	622,3
200.000	200.000	4640,7	913,4	27,3	1288,9
500.000	500.000	15183	2497	77	3546
1.000.000	1.000.000	32799	5351	186	7172



Vremenska složenost funkcija (ms)

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
20	10	0,02231	0,02168	0,02157	0,00532	0,02433	0,02693	0,00428	0,04251
40	20	0,04662	0,04552	0,04419	0,00856	0,04825	0,05138	0,00571	0,08676
100	50	0,11782	0,11447	0,11396	0,01473	0,11963	0,1252	0,01026	0,22035
200	100	0,243	0,2361	0,2337	0,0255	0,244	0,2574	0,0172	0,4494
400	200	0,5011	0,4892	0,4813	0,0465	0,5015	0,5225	0,0322	0,9137
1.000	500	1,2989	1,2669	1,2496	0,1079	1,3113	1,3512	0,076	2,3596
2.000	1.000	2,642	2,581	2,556	0,209	2,633	2,749	0,146	4,849
4.000	2.000	5,4	5,273	5,228	0,426	5,399	5,632	0,315	9,887
10.000	5.000	14,36	14,07	14,12	1,08	14,18	14,73	0,76	25,25
20.000	10.000	28,88	29,02	28,49	2,07	29,26	30,62	1,46	52,28
40.000	20.000	61,05	60,22	59,36	4,21	61,46	64,13	2,92	107,76
100.000	50.000	169	164,4	160,6	10,4	169,8	173,1	7,3	287,9
200.000	100.000	368,5	352,6	349,4	20,9	364,9	375,8	15,3	605,6
400.000	200.000	803,8	761,5	749,9	43,1	809,4	805,1	36,6	1267,4
1.000.000	500.000	2273	2073	2090	109	2246	2236	117	3375
2.000.000	1.000.000	4881	4561	4452	215	4943	4821	243	7147

Vremenska složenost funkcija (ms)

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
25	5	0,01055	0,01061	0,01083	0,00439	0,01275	0,01484	0,00375	0,02176
100	10	0,0211	0,02143	0,02086	0,00612	0,02312	0,02604	0,0053	0,04245
400	20	0,04243	0,04255	0,04192	0,00906	0,04415	0,04918	0,0103	0,08532
2.500	50	0,10776	0,10848	0,10684	0,0148	0,11082	0,12033	0,04022	0,21889
10.000	100	0,2222	0,2226	0,222	0,0262	0,2224	0,2412	0,1302	0,4573
40.000	200	0,4598	0,4835	0,4713	0,0518	0,4744	0,5098	0,4942	0,928
250.000	500	1,162	1,1721	1,1673	0,1571	1,1746	1,2603	3,831	2,3579
1.000.000	1.000	2,36	2,387	2,365	0,353	2,385	2,571	14,666	4,799
4.000.000	2.000	4,771	4,858	4,781	1,001	4,82	5,145	59,439	9,821
25.000.000	5.000	12,59	12,67	12,41	5,86	12,61	13,56	369,08	24,71
100.000.000	10.000	25,09	25,55	25,07	21,02	25,35	27,09	1450	50,62
400.000.000	20.000	53,52	55,34	53,39	89,36	54,05	57,8	6595	106,59
1.600.000.000	40.000	114,7	116,6	114,3	321,5	114,8	127,7	nedovoljna memorija	222,1

t / k

		permutacije			
n	k	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
10	10	0,0030990	0,0027790	0,0003720	0,0042610
20	20	0,0034210	0,0027190	0,0002640	0,0042925
50	50	0,0042360	0,0028348	0,0001836	0,0044664
100	100	0,0046310	0,0028500	0,0001560	0,0045440
200	200	0,0053115	0,0030175	0,0001460	0,0046170
500	500	0,0062708	0,0030678	0,0001374	0,0047366
1.000	1.000	0,0070810	0,0032610	0,0001400	0,0049070
2.000	2.000	0,0080440	0,0033315	0,0001350	0,0050415
5.000	5.000	0,0094160	0,0035360	0,0001340	0,0052340
10.000	10.000	0,0104650	0,0036360	0,0001310	0,0053330
20.000	20.000	0,0121960	0,0038340	0,0001325	0,0055580
50.000	50.000	0,0164100	0,0041440	0,0001380	0,0058800
100.000	100.000	0,0201770	0,0043630	0,0001310	0,0062230
200.000	200.000	0,0232035	0,0045670	0,0001365	0,0064445
500.000	500.000	0,0303660	0,0049940	0,0001540	0,0070920
1.000.000	1.000.000	0,0327990	0,0053510	0,0001860	0,0071720

t / k

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
20	10	0,0022310	0,0021680	0,0021570	0,0005320	0,0024330	0,0026930	0,0004280	0,0042510
40	20	0,0023310	0,0022760	0,0022095	0,0004280	0,0024125	0,0025690	0,0002855	0,0043380
100	50	0,0023564	0,0022894	0,0022792	0,0002946	0,0023926	0,0025040	0,0002052	0,0044070
200	100	0,0024300	0,0023610	0,0023370	0,0002550	0,0024400	0,0025740	0,0001720	0,0044940
400	200	0,0025055	0,0024460	0,0024065	0,0002325	0,0025075	0,0026125	0,0001610	0,0045685
1.000	500	0,0025978	0,0025338	0,0024992	0,0002158	0,0026226	0,0027024	0,0001520	0,0047192
2.000	1.000	0,0026420	0,0025810	0,0025560	0,0002090	0,0026330	0,0027490	0,0001460	0,0048490
4.000	2.000	0,0027000	0,0026365	0,0026140	0,0002130	0,0026995	0,0028160	0,0001575	0,0049435
10.000	5.000	0,0028720	0,0028140	0,0028240	0,0002160	0,0028360	0,0029460	0,0001520	0,0050500
20.000	10.000	0,0028880	0,0029020	0,0028490	0,0002070	0,0029260	0,0030620	0,0001460	0,0052280
40.000	20.000	0,0030525	0,0030110	0,0029680	0,0002105	0,0030730	0,0032065	0,0001460	0,0053880
100.000	50.000	0,0033800	0,0032880	0,0032120	0,0002080	0,0033960	0,0034620	0,0001460	0,0057580
200.000	100.000	0,0036850	0,0035260	0,0034940	0,0002090	0,0036490	0,0037580	0,0001530	0,0060560
400.000	200.000	0,0040190	0,0038075	0,0037495	0,0002155	0,0040470	0,0040255	0,0001830	0,0063370
1.000.000	500.000	0,0045460	0,0041460	0,0041800	0,0002180	0,0044920	0,0044720	0,0002340	0,0067500
2.000.000	1.000.000	0,0048810	0,0045610	0,0044520	0,0002150	0,0049430	0,0048210	0,0002430	0,0071470

t / k

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
25	5	0,0021100	0,0021220	0,0021660	0,0008780	0,0025500	0,0029680	0,0007500	0,0043520
100	10	0,0021100	0,0021430	0,0020860	0,0006120	0,0023120	0,0026040	0,0005300	0,0042450
400	20	0,0021215	0,0021275	0,0020960	0,0004530	0,0022075	0,0024590	0,0005150	0,0042660
2.500	50	0,0021552	0,0021696	0,0021368	0,0002960	0,0022164	0,0024066	0,0008044	0,0043778
10.000	100	0,0022220	0,0022260	0,0022200	0,0002620	0,0022240	0,0024120	0,0013020	0,0045730
40.000	200	0,0022990	0,0024175	0,0023565	0,0002590	0,0023720	0,0025490	0,0024710	0,0046400
250.000	500	0,0023240	0,0023442	0,0023346	0,0003142	0,0023492	0,0025206	0,0076620	0,0047158
1.000.000	1.000	0,0023600	0,0023870	0,0023650	0,0003530	0,0023850	0,0025710	0,0146660	0,0047990
4.000.000	2.000	0,0023855	0,0024290	0,0023905	0,0005005	0,0024100	0,0025725	0,0297195	0,0049105
25.000.000	5.000	0,0025180	0,0025340	0,0024820	0,0011720	0,0025220	0,0027120	0,0738160	0,0049420
100.000.000	10.000	0,0025090	0,0025550	0,0025070	0,0021020	0,0025350	0,0027090	0,1450000	0,0050620
400.000.000	20.000	0,0026760	0,0027670	0,0026695	0,0044680	0,0027025	0,0028900	0,3297500	0,0053295
1.600.000.000	40.000	0,0028675	0,0029150	0,0028575	0,0080375	0,0028700	0,0031925	#VALUE!	0,0055525

t / (k ln k)

		permutacije			
n	k	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
10	10	0,00134588	0,00120690	0,00016156	0,00185053
20	20	0,00114196	0,00090762	0,00008813	0,00143287
50	50	0,00108282	0,00072464	0,00004693	0,00114171
100	100	0,00100561	0,00061887	0,00003387	0,00098672
200	200	0,00100249	0,00056952	0,00002756	0,00087141
500	500	0,00100904	0,00049364	0,00002211	0,00076217
1.000	1.000	0,00102508	0,00047208	0,00002027	0,00071036
2.000	2.000	0,00105830	0,00043830	0,00001776	0,00066328
5.000	5.000	0,00110553	0,00041516	0,00001573	0,00061452
10.000	10.000	0,00113622	0,00039477	0,00001422	0,00057902
20.000	20.000	0,00123149	0,00038714	0,00001338	0,00056122
50.000	50.000	0,00151667	0,00038300	0,00001275	0,00054345
100.000	100.000	0,00175255	0,00037897	0,00001138	0,00054052
200.000	200.000	0,00190098	0,00037416	0,00001118	0,00052797
500.000	500.000	0,00231406	0,00038057	0,00001174	0,00054045
1.000.000	1.000.000	0,00237407	0,00038732	0,00001346	0,00051913

t / (k ln k)

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
20	10	0,00096891	0,00094155	0,00093677	0,00023104	0,00105664	0,00116956	0,00018588	0,00184619
40	20	0,00077811	0,00075975	0,00073755	0,00014287	0,00080531	0,00085755	0,00009530	0,00144806
100	50	0,00060235	0,00058522	0,00058261	0,00007531	0,00061160	0,00064008	0,00005245	0,00112653
200	100	0,00052767	0,00051268	0,00050747	0,00005537	0,00052984	0,00055894	0,00003735	0,00097586
400	200	0,00047289	0,00046166	0,00045420	0,00004388	0,00047326	0,00049308	0,00003039	0,00086225
1.000	500	0,00041802	0,00040772	0,00040215	0,00003472	0,00042201	0,00043485	0,00002446	0,00075937
2.000	1.000	0,00038247	0,00037364	0,00037002	0,00003026	0,00038117	0,00039796	0,00002114	0,00070196
4.000	2.000	0,00035522	0,00034687	0,00034391	0,00002802	0,00035516	0,00037048	0,00002072	0,00065038
10.000	5.000	0,00033720	0,00033039	0,00033156	0,00002536	0,00033297	0,00034589	0,00001785	0,00059292
20.000	10.000	0,00031356	0,00031508	0,00030933	0,00002247	0,00031769	0,00033245	0,00001585	0,00056762
40.000	20.000	0,00030822	0,00030403	0,00029969	0,00002126	0,00031029	0,00032377	0,00001474	0,00054405
100.000	50.000	0,00031239	0,00030389	0,00029686	0,00001922	0,00031387	0,00031997	0,00001349	0,00053217
200.000	100.000	0,00032008	0,00030626	0,00030348	0,00001815	0,00031695	0,00032642	0,00001329	0,00052602
400.000	200.000	0,00032926	0,00031193	0,00030718	0,00001766	0,00033156	0,00032979	0,00001499	0,00051917
1.000.000	500.000	0,00034643	0,00031595	0,00031854	0,00001661	0,00034232	0,00034079	0,00001783	0,00051439
2.000.000	1.000.000	0,00035330	0,00033014	0,00032225	0,00001556	0,00035779	0,00034896	0,00001759	0,00051732

t / (k ln k)

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
25	5	0,00131102	0,00131847	0,00134581	0,00054553	0,00158440	0,00184412	0,00046600	0,00270405
100	10	0,00091636	0,00093069	0,00090594	0,00026579	0,00100409	0,00113090	0,00023018	0,00184358
400	20	0,00070817	0,00071018	0,00069966	0,00015122	0,00073688	0,00082083	0,00017191	0,00142403
2.500	50	0,00055092	0,00055460	0,00054621	0,00007566	0,00056656	0,00061518	0,00020562	0,00111906
10.000	100	0,00048250	0,00048337	0,00048207	0,00005689	0,00048294	0,00052376	0,00028273	0,00099301
40.000	200	0,00043391	0,00045628	0,00044476	0,00004888	0,00044769	0,00048110	0,00046637	0,00087575
250.000	500	0,00037396	0,00037721	0,00037566	0,00005056	0,00037801	0,00040559	0,00123290	0,00075883
1.000.000	1.000	0,00034164	0,00034555	0,00034237	0,00005110	0,00034526	0,00037219	0,00212312	0,00069473
4.000.000	2.000	0,00031384	0,00031957	0,00031450	0,00006585	0,00031707	0,00033845	0,00391000	0,00064604
25.000.000	5.000	0,00029564	0,00029752	0,00029141	0,00013760	0,00029611	0,00031841	0,00866670	0,00058024
100.000.000	10.000	0,00027241	0,00027741	0,00027219	0,00022822	0,00027523	0,00029413	0,01574317	0,00054960
400.000.000	20.000	0,00027021	0,00027940	0,00026955	0,00045115	0,00027288	0,00029182	0,03329635	0,00053814
1.600.000.000	40.000	0,00027060	0,00027509	0,00026966	0,00075850	0,00027084	0,00030127	#VALUE!	0,00052399

$t / (k \ln^2 k)$

		permutacije			
n	k	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
10	10	0,00058451	0,00052415	0,00007016	0,00080367
20	20	0,00038119	0,00030297	0,00002942	0,00047830
50	50	0,00027679	0,00018523	0,00001200	0,00029185
100	100	0,00021837	0,00013439	0,00000736	0,00021426
200	200	0,00018921	0,00010749	0,00000520	0,00016447
500	500	0,00016237	0,00007943	0,00000356	0,00012264
1.000	1.000	0,00014840	0,00006834	0,00000293	0,00010284
2.000	2.000	0,00013923	0,00005766	0,00000234	0,00008726
5.000	5.000	0,00012980	0,00004874	0,00000185	0,00007215
10.000	10.000	0,00012336	0,00004286	0,00000154	0,00006287
20.000	20.000	0,00012435	0,00003909	0,00000135	0,00005667
50.000	50.000	0,00014018	0,00003540	0,00000118	0,00005023
100.000	100.000	0,00015222	0,00003292	0,00000099	0,00004695
200.000	200.000	0,00015574	0,00003065	0,00000092	0,00004326
500.000	500.000	0,00017635	0,00002900	0,00000089	0,00004119
1.000.000	1.000.000	0,00017184	0,00002804	0,00000097	0,00003758

$t / (k \ln^2 k)$

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
20	10	0,00042079	0,00040891	0,00040684	0,00010034	0,00045889	0,00050793	0,00008073	0,00080179
40	20	0,00025974	0,00025361	0,00024620	0,00004769	0,00026882	0,00028626	0,00003181	0,00048337
100	50	0,00015397	0,00014960	0,00014893	0,00001925	0,00015634	0,00016362	0,00001341	0,00028797
200	100	0,00011458	0,00011133	0,00011020	0,00001202	0,00011505	0,00012137	0,00000811	0,00021191
400	200	0,00008925	0,00008713	0,00008573	0,00000828	0,00008932	0,00009306	0,00000574	0,00016274
1.000	500	0,00006726	0,00006561	0,00006471	0,00000559	0,00006791	0,00006997	0,00000394	0,00012219
2.000	1.000	0,00005537	0,00005409	0,00005357	0,00000438	0,00005518	0,00005761	0,00000306	0,00010162
4.000	2.000	0,00004673	0,00004563	0,00004525	0,00000369	0,00004673	0,00004874	0,00000273	0,00008557
10.000	5.000	0,00003959	0,00003879	0,00003893	0,00000298	0,00003909	0,00004061	0,00000210	0,00006961
20.000	10.000	0,00003404	0,00003421	0,00003358	0,00000244	0,00003449	0,00003610	0,00000172	0,00006163
40.000	20.000	0,00003112	0,00003070	0,00003026	0,00000215	0,00003133	0,00003269	0,00000149	0,00005494
100.000	50.000	0,00002887	0,00002809	0,00002744	0,00000178	0,00002901	0,00002957	0,00000125	0,00004919
200.000	100.000	0,00002780	0,00002660	0,00002636	0,00000158	0,00002753	0,00002835	0,00000115	0,00004569
400.000	200.000	0,00002698	0,00002556	0,00002517	0,00000145	0,00002716	0,00002702	0,00000123	0,00004253
1.000.000	500.000	0,00002640	0,00002408	0,00002427	0,00000127	0,00002609	0,00002597	0,00000136	0,00003920
2.000.000	1.000.000	0,00002557	0,00002390	0,00002332	0,00000113	0,00002590	0,00002526	0,00000127	0,00003744

$t / (k \ln^2 k)$

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
25	5	0,00081458	0,00081921	0,00083620	0,00033896	0,00098445	0,00114582	0,00028954	0,00168012
100	10	0,00039797	0,00040419	0,00039344	0,00011543	0,00043607	0,00049114	0,00009996	0,00080066
400	20	0,00023639	0,00023706	0,00023355	0,00005048	0,00024598	0,00027400	0,00005739	0,00047535
2.500	50	0,00014083	0,00014177	0,00013962	0,00001934	0,00014483	0,00015725	0,00005256	0,00028606
10.000	100	0,00010477	0,00010496	0,00010468	0,00001235	0,00010487	0,00011373	0,00006139	0,00021563
40.000	200	0,00008190	0,00008612	0,00008394	0,00000923	0,00008450	0,00009080	0,00008802	0,00016529
250.000	500	0,00006017	0,00006070	0,00006045	0,00000814	0,00006083	0,00006526	0,00019839	0,00012210
1.000.000	1.000	0,00004946	0,00005002	0,00004956	0,00000740	0,00004998	0,00005388	0,00030735	0,00010057
4.000.000	2.000	0,00004129	0,00004204	0,00004138	0,00000866	0,00004171	0,00004453	0,00051441	0,00008500
25.000.000	5.000	0,00003471	0,00003493	0,00003421	0,00001616	0,00003477	0,00003738	0,00101755	0,00006813
100.000.000	10.000	0,00002958	0,00003012	0,00002955	0,00002478	0,00002988	0,00003193	0,00170929	0,00005967
400.000.000	20.000	0,00002728	0,00002821	0,00002722	0,00004556	0,00002755	0,00002947	0,00336208	0,00005434
1.600.000.000	40.000	0,00002554	0,00002596	0,00002545	0,00007158	0,00002556	0,00002843	#VALUE!	0,00004945

t / n

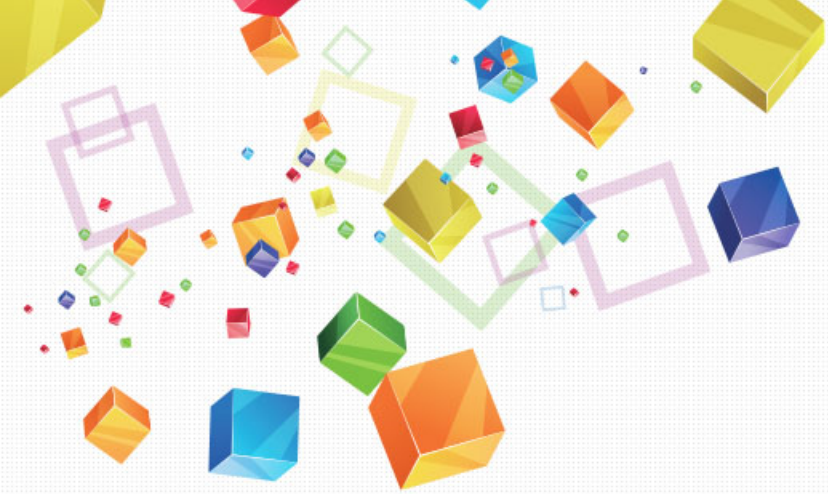
		permutacije			
n	k	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
10	10	0,0030990	0,0027790	0,0003720	0,0042610
20	20	0,0034210	0,0027190	0,0002640	0,0042925
50	50	0,0042360	0,0028348	0,0001836	0,0044664
100	100	0,0046310	0,0028500	0,0001560	0,0045440
200	200	0,0053115	0,0030175	0,0001460	0,0046170
500	500	0,0062708	0,0030678	0,0001374	0,0047366
1.000	1.000	0,0070810	0,0032610	0,0001400	0,0049070
2.000	2.000	0,0080440	0,0033315	0,0001350	0,0050415
5.000	5.000	0,0094160	0,0035360	0,0001340	0,0052340
10.000	10.000	0,0104650	0,0036360	0,0001310	0,0053330
20.000	20.000	0,0121960	0,0038340	0,0001325	0,0055580
50.000	50.000	0,0164100	0,0041440	0,0001380	0,0058800
100.000	100.000	0,0201770	0,0043630	0,0001310	0,0062230
200.000	200.000	0,0232035	0,0045670	0,0001365	0,0064445
500.000	500.000	0,0303660	0,0049940	0,0001540	0,0070920
1.000.000	1.000.000	0,0327990	0,0053510	0,0001860	0,0071720

t / n

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
20	10	0,0011155	0,0010840	0,0010785	0,0002660	0,0012165	0,0013465	0,0002140	0,0021255
40	20	0,0011655	0,0011380	0,0011048	0,0002140	0,0012063	0,0012845	0,0001428	0,0021690
100	50	0,0011782	0,0011447	0,0011396	0,0001473	0,0011963	0,0012520	0,0001026	0,0022035
200	100	0,0012150	0,0011805	0,0011685	0,0001275	0,0012200	0,0012870	0,0000860	0,0022470
400	200	0,0012528	0,0012230	0,0012033	0,0001163	0,0012538	0,0013063	0,0000805	0,0022843
1.000	500	0,0012989	0,0012669	0,0012496	0,0001079	0,0013113	0,0013512	0,0000760	0,0023596
2.000	1.000	0,0013210	0,0012905	0,0012780	0,0001045	0,0013165	0,0013745	0,0000730	0,0024245
4.000	2.000	0,0013500	0,0013183	0,0013070	0,0001065	0,0013498	0,0014080	0,0000788	0,0024718
10.000	5.000	0,0014360	0,0014070	0,0014120	0,0001080	0,0014180	0,0014730	0,0000760	0,0025250
20.000	10.000	0,0014440	0,0014510	0,0014245	0,0001035	0,0014630	0,0015310	0,0000730	0,0026140
40.000	20.000	0,0015263	0,0015055	0,0014840	0,0001053	0,0015365	0,0016033	0,0000730	0,0026940
100.000	50.000	0,0016900	0,0016440	0,0016060	0,0001040	0,0016980	0,0017310	0,0000730	0,0028790
200.000	100.000	0,0018425	0,0017630	0,0017470	0,0001045	0,0018245	0,0018790	0,0000765	0,0030280
400.000	200.000	0,0020095	0,0019038	0,0018748	0,0001078	0,0020235	0,0020128	0,0000915	0,0031685
1.000.000	500.000	0,0022730	0,0020730	0,0020900	0,0001090	0,0022460	0,0022360	0,0001170	0,0033750
2.000.000	1.000.000	0,0024405	0,0022805	0,0022260	0,0001075	0,0024715	0,0024105	0,0001215	0,0035735

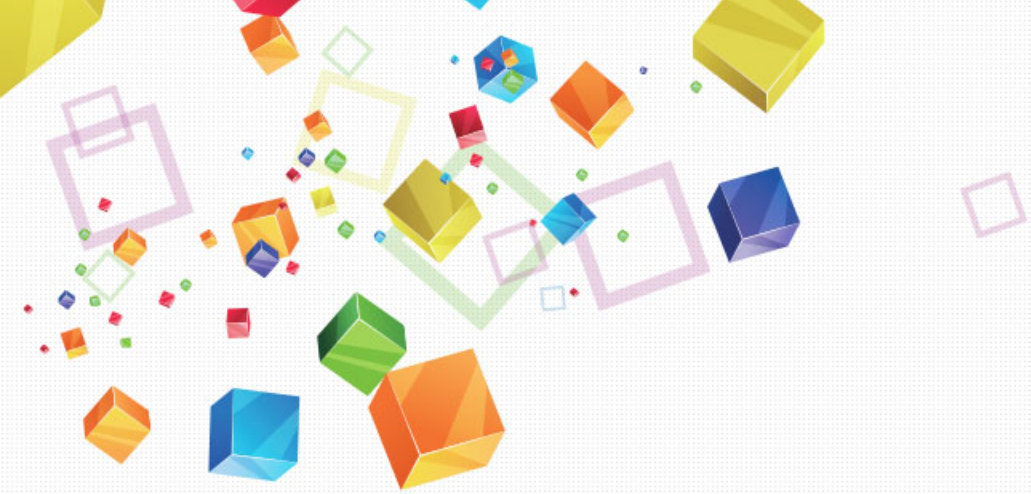
t / n

		podskupovi				permutacije			
n	k	Algoritam pokušaja	Floydov algoritam, rekurzivno	Floydov algoritam, nerekurzivno	Floydov algoritam, poljem bitova	Algoritam pokušaja	Generiranje podskupa (Floyd) i miješanje	Algoritam miješanja	Floydov algoritam
25	5	0,00042200	0,00042440	0,00043320	0,00017560	0,00051000	0,00059360	0,00015000	0,00087040
100	10	0,00021100	0,00021430	0,00020860	0,00006120	0,00023120	0,00026040	0,00005300	0,00042450
400	20	0,00010608	0,00010638	0,00010480	0,00002265	0,00011038	0,00012295	0,00002575	0,00021330
2.500	50	0,00004310	0,00004339	0,00004274	0,00000592	0,00004433	0,00004813	0,00001609	0,00008756
10.000	100	0,00002222	0,00002226	0,00002220	0,00000262	0,00002224	0,00002412	0,00001302	0,00004573
40.000	200	0,00001150	0,00001209	0,00001178	0,00000130	0,00001186	0,00001275	0,00001236	0,00002320
250.000	500	0,00000465	0,00000469	0,00000467	0,00000063	0,00000470	0,00000504	0,00001532	0,00000943
1.000.000	1.000	0,00000236	0,00000239	0,00000237	0,00000035	0,00000239	0,00000257	0,00001467	0,00000480
4.000.000	2.000	0,00000119	0,00000121	0,00000120	0,00000025	0,00000121	0,00000129	0,00001486	0,00000246
25.000.000	5.000	0,00000050	0,00000051	0,00000050	0,00000023	0,00000050	0,00000054	0,00001476	0,00000099
100.000.000	10.000	0,00000025	0,00000026	0,00000025	0,00000021	0,00000025	0,00000027	0,00001450	0,00000051
400.000.000	20.000	0,00000013	0,00000014	0,00000013	0,00000022	0,00000014	0,00000014	0,00001649	0,00000027
1.600.000.000	40.000	0,00000007	0,00000007	0,00000007	0,00000020	0,00000007	0,00000008	#VALUE!	0,00000014



Literatura

- Jon Bentley, *More Programming Pearls*, Addison-Wesley publishing company, 1988.
- https://channel9.msdn.com/Events/GoingNative/2013/rand-Considered-Harmful , 6. siječnja 2017.



Hvala na pažnji!