

Generacja (n,k)-kombinacji - algorytmy

Sprawozdanie z laboratorium 1 – Piotr Sarna LK1

Cel ćwiczenia

Podczas zajęć zapoznaliśmy się z dwoma algorytmami, mającymi na celu wygenerowanie kombinacji od 1 do zadanego „n”. Były to: algorytm generacji kombinacji w porządku leksykograficznym oraz algorytm Semby.

Wstęp teoretyczny

Kombinacja to „k”-elementowy podzbiór skończonego „n”-elementowego zbioru, gdzie $0 \leq k \leq n$.

Liczbę kombinacji możemy wyliczyć za pomocą współczynnika dwumianowego Newtona:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Źródło: https://pl.wikipedia.org/wiki/Symbol_Newtona

Porządek leksykograficzny oznacza, że cyfry w kombinacji będą uporządkowane od najmniejszej do największej.

Opis algorytmów

1. Algorytm generujący kombinacje w porządku leksykograficznym

Algorytm ten dla podanych wartości „n” i „k” (gdzie „n” jest wielkością zbioru od 1 do „n”, a „k” jest ilością elementów w generowanej kombinacji) tworzy kombinacje, w których każda kolejna cyfra jest większa od poprzedniej.

Zapis algorytmu w pseudokodzie:

1. Dla $i = 1$ do k wykonaj
 - 1.1. $K[i] = i; L[i] = n - k + i;$
2. Powtarzaj
 - 2.1. $i = k;$
 - 2.2. Jeżeli $K[i] < L[i]$ to
 - 2.2.1. $K[i] = K[i] + 1;$
 - inaczej
 - 2.2.2. Powtarzaj $i = i - 1$ dopóki $K[i] < L[i];$
 - 2.2.3. $K[i] = K[i] + 1;$
 - 2.2.4. Dla $j = i + 1$ do k wykonaj $K[j] = K[j - 1] + 1;$
 - 2.3. Wyprowadź $K[1] \dots K[k];$
dopóki $K[1] = L[1];$

Źródło: materiały Z.Kokosiński

Prezentacja działania mojej implementacji w C++

```
Konsola debugowania programu Microsoft Visual Studio
Generacja (n,k)-kombinacji - algorytmy
1. Algorytm generacji kombinacji w rosnacym porzadku leksykograficznym
2. Algorytm Semby
Wybierz algorytm:
1
Podaj wielkosc zbioru:
6
Podaj wielkosc podzbioru:
4
Wynik zostal zapisany do pliku wynik.txt
Czas dzialania algorytmu wynosi 702 mikrosekund
D:\Piotr\Studia\2_semestr\MetodyProgramowania\L1\x64\Debug\L1.exe (proc
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opc
znie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

wynik.txt	L1.cpp
1	1 2 3 4
2	1 2 3 5
3	1 2 3 6
4	1 2 4 5
5	1 2 4 6
6	1 2 5 6
7	1 3 4 5
8	1 3 4 6
9	1 3 5 6
10	1 4 5 6
11	2 3 4 5
12	2 3 4 6
13	2 3 5 6
14	2 4 5 6
15	3 4 5 6

n	k	Czas pracy algorytmu
2	1	298 μ s
4	2	418 μ s
6	4	702 μ s
8	6	1132 μ s
10	8	2274 μ s

2. Algorytm Semby

Algorytm ten służy do generowania wszystkich możliwych ciągów liczb rosnących z liczb zawartych w zbiorze od 1 do „n”.

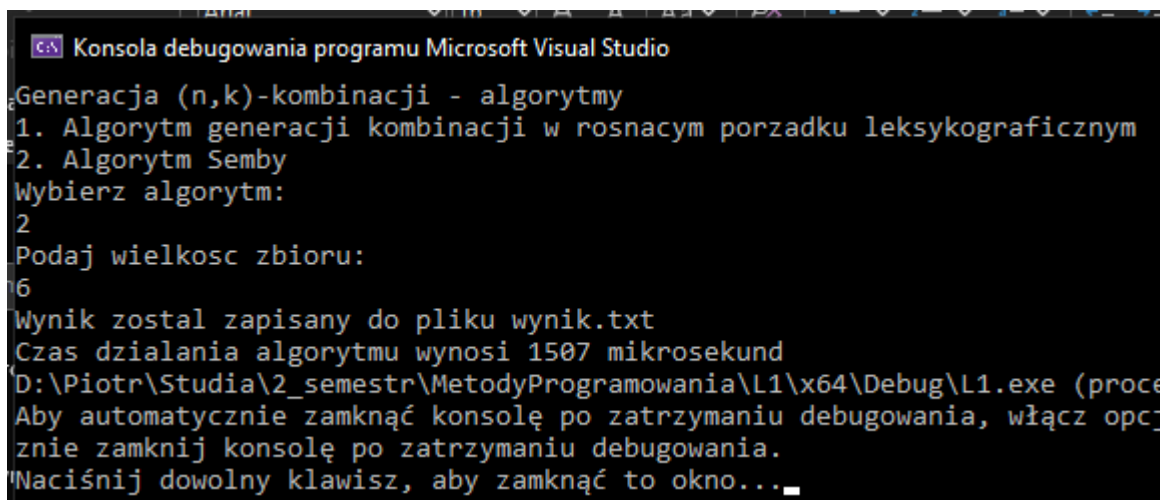
W wyniku jego działania powstaje $2^n - 1$ ciągów, co już dla stosunkowo niewielkich zbiorów zawierających np. 10 elementów tworzy 1023 podzbiory. Algorytm posiada złożoność czasową wykładniczą.

Zapis algorytmu w pseudokodzie:

1. $k = 0; K[k] = 0;$
 2. Powtarzaj
 - 2.1. Jeżeli $K[k] < n$, to *extend* inaczej *reduce*;
 - 2.2. Wyprowadź $K[1] \dots K[k];$
dopóki $K[1] = n;$
- extend* $K[k + 1] = K[k] + 1; k = k + 1;$
reduce $k = k - 1; K[k] = K[k] + 1;$

Źródło: materiały Z.Kokosiński

Prezentacja działania mojej implementacji w C++



```
Konsola debugowania programu Microsoft Visual Studio
Generacja (n,k)-kombinacji - algorytmy
1. Algorytm generacji kombinacji w rosnacym porzadku leksykograficznym
2. Algorytm Semby
Wybierz algorytm:
2
Podaj wielkosc zbioru:
6
Wynik zostal zapisany do pliku wynik.txt
Czas dzialania algorytmu wynosi 1507 mikrosekund
D:\Piotr\Studia\2_semestr\MetodyProgramowania\L1\x64\Debug\L1.exe (proces)
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję
znie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

```

1 1
2 1 2
3 1 2 3
4 1 2 3 4
5 1 2 3 4 5
6 1 2 3 4 5 6
7 1 2 3 4 6
8 1 2 3 5
9 1 2 3 5 6
10 1 2 3 6
11 1 2 4
12 1 2 4 5
13 1 2 4 5 6
14 1 2 4 6
15 1 2 5
16 1 2 5 6
17 1 2 6
18 1 3
19 1 3 4
20 1 3 4 5
21 1 3 4 5 6
22 1 3 4 6
23 1 3 5
24 1 3 5 6
25 1 3 6
26 1 4
27 1 4 5
28 1 4 5 6
29 1 4 6
30 1 5
31 1 5 6
32 1 6
33 2
34 2 3
35 2 3 4
36 2 3 4 5
37 2 3 4 5 6
38 2 3 4 6
39 2 3 5
40 2 3 5 6
41 2 3 6
42 2 4
43 2 4 5
44 2 4 5 6
45 2 4 6
46 2 5
47 2 5 6
48 2 6
49 3
50 3 4
51 3 4 5
52 3 4 5 6
53 3 4 6
54 3 5
55 3 5 6
56 3 6
57 4
58 4 5
59 4 5 6
60 4 6
61 5
62 5 6
63 6

```

n	Czas pracy algorytmu
2	474 μ s
4	517 μ s
6	1507 μ s
8	8521 μ s
10	39743 μ s

Wnioski

Przedstawione algorytmy służą do wygenerowania kombinacji od 1 do „n”, lecz mają różne właściwości i zastosowania.

Pierwszy algorytm pozwolił nam utworzyć wszystkie „k” elementowe kombinacje zadanego zbioru w rosnącej kolejności, natomiast drugi utworzył **wszystkie** rosnące kombinacje zadanego zbioru.

Algorytm Semby posiada wykładniczą złożoność obliczeniową

Bibliografia

<https://cplusplus.com/doc/tutorial/files/>

<https://www.geeksforgeeks.org/measure-execution-time-function-cpp/>