Zadanie 1 - przykład.

W przykładzie używam kodów bitowych, ale można też operować bezpośrednio na podzbiorach.

Niech n=4. Pierwszy kod składa się zawsze z samych zer:

0000 (zatem wypisujemy zbiór pusty).

Kolejne kody otrzymujemy, wykorzystując algorytm następnika. Jeśli waga Hamminga (czyli liczba jedynek) bieżącego kodu jest parzysta – tak jak dla kodu 0000, to wystarczy zmienić ostatni bit, otrzymujemy kolejny kod:

0001 (czyli podzbiór {4}).

Jeśli waga Hamminga bieżącego kodu jest nieparzysta – tak jak dla kodu 0001, to szukamy od prawej strony pierwszej jedynki i zmieniamy bit na lewo od niej:

0011 (podzbiór {3,4}).

W kolejnym kroku znów mamy parzystą wagę Hamminga, więc zmieniamy ostatni bit:

0010 (podzbiór {3}).

W następnym kroku ponownie waga Hamminga jest nieparzysta, zmieniamy bit poprzedzający ostatnią jedynkę:

0110 (podzbiór {2,3}).

I tak dalej, aż otrzymamy kod składający się z jedynki na początku i samych zer:

1000 (podzbiór {1}).

W kolejnym kroku szukając bitu do zmiany wyjdziemy przed tablicę z naszym ciągiem – to oznacza, że nie istnieje następnik bieżącego kodu, czyli wygenerowaliśmy już wszystkie podzbiory.

Zadanie 2 - przykład.

Niech n=4, T={1,3}.

Zamieniamy podzbiór T na jego kod bitowy: 1010. Obliczamy kolejne bity rangi podzbioru, xor-ując poprzedni bit rangi z odpowiednim bitem kodu podzbioru. Wartość początkowa poprzedniego bitu rangi to 0.

	1	0	1	0
0				

	1	0	1	0
0	1			

	1	0	1	0
0	1	1		

	1	0	1	0
0	1	1	0	

	1	0	1	0
0	1	1	0	0

Ranga bitowo to 1100, po zamianie na system dziesiętny dostajemy odpowiedź: r=12.

Zadanie 3 - przykład.

Mając daną rangę r, kod bitowy podzbioru otrzymujemy ze wzoru: r xor (r div 2).

Niech n=4, r=12. Zapisujemy rangę bitowo: 1100, r div 2 to 110.

r	1	1	0	0
r div 2	0	1	1	0
xor	1	0	1	0

Otrzymany kod bitowy 1010 zamieniamy na podzbiór, otrzymujemy T={1,3}.