**SINH KẾ TIẾP**

**\* Khái quát:**

- **Ý tưởng** của phương pháp sinh là từ cấu hình đầu tiên  ta sinh ra các cấu hình tiếp theo theo một cách nào đó. Tuy nhiên không phải bài nào cũng có thể làm theo phương pháp sinh.

- **Yêu cầu** của bài toán để có thể làm theo phương pháp sinh là:

+ **Thứ nhất** phải xác định được cấu hình đầu và cấu hình cuối  
+ **Thứ hai** phải xác định được cấu hình tiếp theo bằng 1 công thức nhất định

1. **Sinh nhị phân:** Liệt kê tất cả xâu nhị phân n bit
2. **Nguyên lí hoạt động**

**\* *Cấu hình ban đầu***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**\**Cấu hình cuối***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **1** | **1** | **1** | **1** | **1** | **1** |

*\** ***Cấu hình hiện tại***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **1** | **0** | **1** | **0** | **1** | **1** |

***\* Cấu hình kế tiếp***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **1** | **0** | **1** | **1** | **0** | **0** |

Cấu hình này được sinh ra từ cấu hình trước đó bằng cách cộng 1 (từ phải sang trái)

**=> Ý tưởng** sinh xâu nhị phân kế tiếp A = b1b2…bn (b # 11…1) là cộng 1 (từ phải sang trái) vào cấu hình trước (**cong(A,1)**) cụ thể như sau:

1. Tìm từ phải qua trái, chỉ số i đầu tiên mà bit bi = 0
2. Gán lại bi = 1 và các bit ở bên phải gán bj = 0 (j > i)

**=> Thuật toán để giải bài này:**

**B1:** Khởi tạo mảng A gồm n số 0 (tương ứng với chuỗi bit đầu tiên)

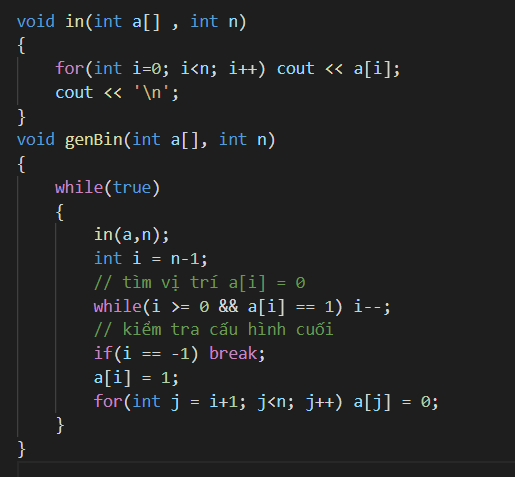
**B2:** In cấu hình hiện tại

**B3:**

**-** Check nếu là cấu hình cuối thì break;

- Ngược lại **cong(A,1)**, lặp lại từ bước **B2**

**\* Code:**



1. **Độ phức tạp**

**-** a[i] nhận giá trị 0 hoặc 1

**-** duyệt n lần

=> O(2n)

1. **Sinh tổ hợp:** Liệt kê tổ hợp chập K của N phần tử
2. **Nguyên lí hoạt động**

Ví dụ: Liệt kê tập con 4 phần tử của tập {1,2,3,4,5,6}

**\* *Cấu hình ban đầu***

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **2** | **3** | **4** |

**\**Cấu hình cuối***

|  |  |  |  |
| --- | --- | --- | --- |
| **3** | **4** | **5** | **6** |

*\** ***Cấu hình hiện tại***

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **2** | **5** | **6** |

**2 != (6- 4 +2)**

***\* Cấu hình kế tiếp***

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **3** | **4** | **5** |

**\* Thuật toán:**

**B1:** Tạo cấu hình đầu tiên (1,2,…,K)

**B2:** In cấu hình hiện tại

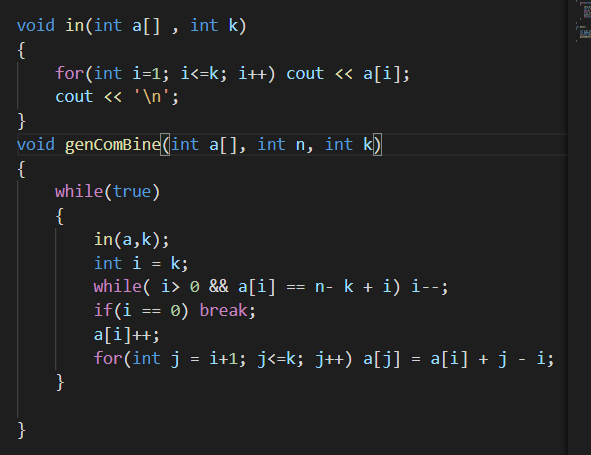
**B3:** Duyệt từ cuối lên, nếu tại vị trí i, Ai!= N - K + i (chưa đạt cấu hình cuối) thì:

- gán Ai = Ai + 1

- update các phần tử sau tăng dần theo thứ tự từ điển tính từ Ai

- quay lại **B2**

\***Code:**



1. **Độ phức tạp**

O(C(k,n))

1. **Sinh hoán vị**
2. **Nguyên lí hoạt động**

**\* *Cấu hình ban đầu***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |

**\**Cấu hình cuối***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **6** | **5** | **4** | **3** | **2** | **1** |

*\** ***Cấu hình hiện tại***

**Giả sử:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | **2** | **6** | **4** | **3** | **1** |

**3 > 2**

***\* Cấu hình kế tiếp***

**Swap(2,3)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | **3** | **6** | **4** | **2** | **1** |

**Sort**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | **3** | **1** | **2** | **4** | **6** |

**\*Thuật toán**

**B1:** Tạo cấu hình đầu tiên (1,2,…,n)

**B2:** In cấu hình

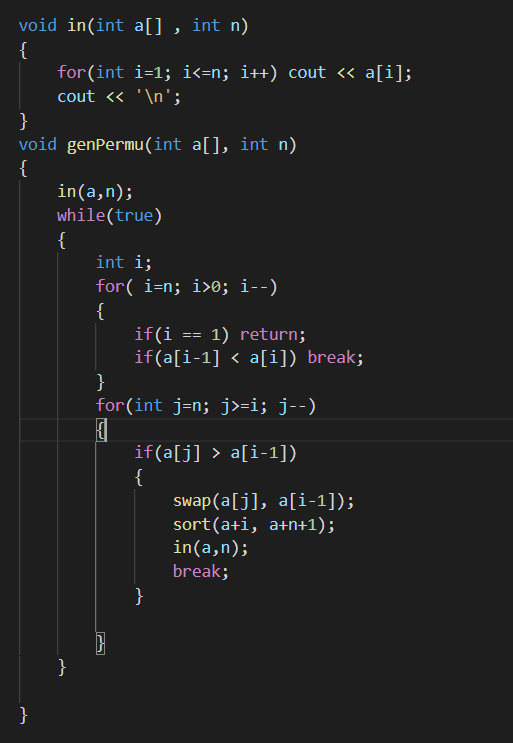
**B3:** - Duyệt từ phải qua trái nếu Ai-1 < Ai thì

+ tìm minAj > Ai (i <= j <= n) , swap(Ai-1, Aj)

+ sắp xếp tăng dần từ Ai đến An

Lặp lại **B2**

**\*Code**



1. **Độ phức tạp**

O(n!)

<https://expressmagazine.net/development/4003/bieu-dien-n-thanh-tong-cac-so-tu-nhien-khong-lon-hon-n-bang-cc> - toán rời rạc

<https://laptrinhsgu.wordpress.com/2019/11/12/bai-toan-chia-keo/>

<https://chithu.violet.vn/entry/show/entry_id/1300644>

<https://simplecodecjava.blogspot.com/2015/08/bai-toan-bieu-dien-n-thanh-tong-cac-so.html>

<https://chuyentincbg.wordpress.com/2012/01/06/ap-d%E1%BB%A5ng-ph%C6%B0%C6%A1ng-phap-sinh-d%E1%BB%83-phan-tich-s%E1%BB%91-nguyen-d%C6%B0%C6%A1ng-n-thanh-t%E1%BB%95ng-cac-s%E1%BB%91-nguyen-d%C6%B0%C6%A1ng/>

<https://ideone.com/gGkYiv>

<https://paste.ofcode.org/3ZRrbPs7nmsRe4673SQuy3>