Chương 7

Quy hoạch động

Quy hoạch động (2)

7.1. Lý thuyết về quy hoạch động

7.2. Bài toán ba lô 1

7.3. Bài toán ba lô 2

7.4. Bài toán dãy con có tổng chia hết cho k

7.3. Bài toán ba lô 2

Cho **n** gói hàng.

Gói hàng thứ **i** có khối lượng là **A[i]** và giá trị **C[i]**.

Cần chọn những gói hàng nào để bỏ vào một ba lô sao tổng giá trị của các gói hàng đã chọn là **lớn nhất** nhưng tổng khối lượng của chúng **không vượt quá** khối lượng **M** cho trước.

Mỗi món hàng có thể lấy **nhiều lần**.

Cho biết số lần chọn từng món hàng

7.3. Bài toán ba lô 2

- Ví dụ: **n = 5; M = 13**

  Tức là cho 5 loại gói hàng. Các gói hàng thứ từ 1 đến 5 lần lượt có khối lượng là **A[i]** và giá trị **C[i]** được cho như trong bảng minh hoạ:

i	1	2	3	4	5
A[i]	3	4	5	2	1
C[i]	4	5	6	3	1

Yêu cầu: Chọn các gói hàng mà tổng giá trị của các gói hàng là lớn nhất nhưng tổng khối lượng của chúng không vượt quá **M = 13**. Mỗi món hang có thể chọn **nhiều lần**

Kết quả: Tổng giá trị của các gói hàng bỏ vào ba lô: **19**

Các gói được chọn: **gói hàng 1 lấy 1 lần**

gói hàng 4 lấy 5 lần

7.3. Bài toán ba lô 2

Tham số thể hiện kích thước bài toán

❖ Kết quả bài toán là tổng giá trị lớn nhất của các món hàng được chọn trong n món sao cho tổng khối lượng không lớn hơn **M** cho trước.

❖ Bài toán có 2 tham số: số lượng món hàng n và khối lượng giới hạn M

7.3. Bài toán ba lô 2

Lập công thức đệ qui

Gọi **F(i, v)** là tổng giá trị lớn nhất của các món hàng được chọn sao cho tổng khối lượng <= v trong i loại hàng.

❖ Trường hợp **A[i] > v**: **F(i, v) = F(i -1, v)**

❖ Trường hợp **A[i] <= v**:

- Nếu gói hàng thứ i không được chọn thì: **F(i, v) = F(i -1, v)**
- Nếu có k gói hàng thứ i được chọn (1 <= k <= v/A[i]) thì:

F(i, v) = Max{ F(i -1, v); F(i -1, v – A[i] * k) + C[i] * k) }

❖ Bài toán nhỏ nhất ứng với **i = 0** ta có: **F(0, v) = 0**

7.3. Bài toán ba lô 2

Lập công thức đệ qui

❖ Với **i = 0** ta có: **F(0, v) = 0**

❖ Với **i > 0**

- Trường hợp **A[i] > v**: **F(i, v) = F(i -1, v)**
- Trường hợp **A[i] <= v**:

F(i, v) = Max{ F(i -1, v); F(i -1, v – A[i] * k) + C[i] * k }

7.3. Bài toán ba lô 2

Xây dựng bảng phương án:

❖ **Cấu trúc bảng phương án:**

- Dùng mảng **F [0..n] [0..M]** chứa giá trị của các **F(i, v)**

- Dùng mảng **S [1..n] [1..M]** chứa giá trị của các **S[i,v]** là số lượng món hàng loại i được chọn

  - Nếu **F(i, v) = F(i - 1, v)**: **S[i, v] = 0**

  - Nếu **F(i, v) ≠ F(i - 1, v)**: **S[i, v] = k**

7.3. Bài toán ba lô 2

Xây dựng bảng phương án:

❖ **Cách tính giá trị trên bảng phương án:**

Điền số **0** cho các ô trên dòng **0** của bảng phương án **F**

Sử dụng công thức đệ qui và giá trị trên dòng **(i -1)** để tính dòng **i** của bảng **F** và bảng **S**

- Trường hợp **A[i] > v**: **F(i, v) = F(i -1, v)**

- Trường hợp **A[i] <= v**:

F(i, v) = Max{ F(i -1, v); F(i -1, v – A[i] * k) + C[i] * k }

Với **k** là số gói hàng thứ **i** được chọn (1 <= k <= v/A[i])

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 0

Điền số 0 cho các ô trên dòng 0 của bảng phương án F

Bảng F[0, v]

C	A	v \ i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 1

Bảng F[1, v]

C	A	v / i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	**1**	0	0	0	4	4	4	8	8	8	12	12	12	16	16

Bảng S[1, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	**1**		0	0	1	1	1	2	2	2	3	3	3	4	4

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 2

Bảng F[2, v]

C	A	v / i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	**1**	0	0	0	4	4	4	8	8	8	12	12	12	16	16
5	4	**2**	0	0	0	4	5	5	8	9	10	12	13	14	16	17

Bảng S[2, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	**1**		0	0	1	1	1	2	2	2	3	3	3	4	4
5	4	**2**		0	0	0	1	1	0	1	2	0	1	2	0	1

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 3

Bảng F[3, v]

C	A	v / i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	**1**	0	0	0	4	4	4	8	8	8	12	12	12	16	16
5	4	**2**	0	0	0	4	5	5	8	9	10	12	13	14	16	17
6	5	**3**	0	0	0	4	5	6	8	9	10	12	13	14	16	17

Bảng S[3, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	**1**		0	0	1	1	1	2	2	2	3	3	3	4	4
5	4	**2**		0	0	0	1	1	0	1	2	0	1	2	0	1
6	5	**3**		0	0	0	0	1	0	0	0	0	0	0	0	0

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 4

Bảng F[4, v]

C	A	v / i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	1	0	0	0	4	4	4	8	8	8	12	12	12	16	16
5	4	2	0	0	0	4	5	5	8	9	10	12	13	14	16	17
6	5	3	0	0	0	4	5	6	8	9	10	12	13	14	16	17
3	2	4	0	0	3	4	6	7	9	10	12	13	15	16	18	19

Bảng S[4, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	1		0	0	1	1	1	2	2	2	3	3	3	4	4
5	4	2		0	0	0	1	1	0	1	2	0	1	2	0	1
6	5	3		0	0	0	0	0	1	0	0	0	0	0	0	0
3	2	4		0	1	0	2	1	3	2	4	3	5	4	6	5

7.3. Bài toán ba lô 2

Ví dụ bảng phương án: Với i = 5

Bảng F[5, v]

C	A	v / i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	**1**	0	0	0	4	4	4	8	8	8	12	12	12	16	16
5	4	**2**	0	0	0	4	5	5	8	9	10	12	13	14	16	17
6	5	**3**	0	0	0	4	4	5	8	9	10	12	13	14	16	17
3	2	**4**	0	0	3	4	6	7	9	10	12	13	15	16	18	19
1	1	**5**	0	1	3	4	6	7	9	10	12	13	15	16	18	**19**

Bảng S[5, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	**1**		0	0	1	1	1	2	2	2	3	3	3	4	4
5	4	**2**		0	0	0	0	1	0	1	1	0	1	2	0	1
6	5	**3**		0	0	0	0	0	0	0	1	0	0	0	0	0
3	2	**4**		0	1	0	2	1	3	2	4	3	5	4	6	5
1	1	**5**		1	0	0	0	0	0	0	0	0	0	0	0	**0**

7.3. Bài toán ba lô 2

Thuật toán tạo bảng phương án

```
void TaoBangPhuongAn(F[0..n] [0..M], S[1..n] [1..M] )
{  for (v=0; v <= M; v++)
        F[0, v] = 0; // Dòng 0 bảng F[i, v]
   for (i = 1; i <= n; i++)
      for (v=0; v <= M; v++)
      {  F[i, v] = F[i-1, v];                S[i, v] = 0;
         if (v >= A[i])
         for(k = 1; k <= v/ A[i]; k++)
            if (F[i, v]  <  F[i-1, v - A[i]*k ] + C[i]*k)
               { F[i, v] = F[i-1, v - A[i] *k ] + C[i] *k;
                S[i, v] = k;
               }
      }
}
```

7.3. Bài toán ba lô 2

Truy vết tìm lại các gói hàng đã chọn:

Bắt đầu từ ô **S[n, M]** trên dòng n ta dò ngược về dòng **1** theo nguyên tắc:

❖ Nếu **S[i, v] <> 0** thì:

- Loại hàng **i** được chọn với số lượng là **S[i, v]**
- Truy tiếp ô **S[i-1, v - S[i, v]*A[i]]**.

❖ Nếu **S[i, v] = 0** thì:

- Loại hàng **i** không được chọn.
- Truy tiếp ô **S[i-1, v]**.

7.3. Bài toán ba lô 2

Truy vết tìm lại các gói hàng đã chọn:

Bảng S[i, v]

C	A	v / i		1	2	3	4	5	6	7	8	9	10	11	12	13
4	3	**1**		0	0	**1**	1	1	2	2	2	3	3	3	4	4
5	4	**2**		0	0	**0**	1	1	0	1	2	0	1	2	0	1
6	5	**3**		0	0	**0**	0	0	0	0	1	0	0	0	0	0
3	2	**4**		0	1	0	2	1	3	2	4	3	5	4	6	**5**
1	1	**5**		1	0	0	0	0	0	0	0	0	0	0	0	**0**

7.3. Bài toán ba lô 2

Thuật toán truy vết tìm lại các gói hàng đã chọn

```
void TruyVet(S [1..n] [1..M])
{  Bắt đầu từ ô S[n, M] trên dòng n: i = n; v = M;

   while (i > 0)

   {  if (S[i, v] != 0)

        {  <in số lượng gói hàng i trong S[i, v] >;

           v = v - S[i, v]*A[i];
        }
     i = i -1;
   }
}
```

Quy hoạch động (2)

7.1. Lý thuyết về quy hoạch động

7.2. Bài toán ba lô 1

7.3. Bài toán ba lô 2

7.4. Bài toán dãy con có tổng chia hết cho k

7.4. Bài toán dãy con có tổng chia hết cho k

Bài toán:

Cho một dãy **A** gồm **n** số nguyên.

Cho **k** là một số nguyên dương.

Hãy tìm một dãy con (không nhất thiết phải liên tiếp nhau) dài nhất có tổng các số chia hết cho số **k**.

7.4. Bài toán dãy con có tổng chia hết cho k

- Ví dụ: **n = 6; k = 5**

  Tức là dãy gồm 6 số nguyên dương. Số thứ tự các số trong dãy và giá trị của chúng lần lượt được cho tương ứng như trong bảng minh hoạ:

i	1	2	3	4	5	6
A[i]	**11**	**6**	**7**	**12**	**20**	**8**

Yêu cầu: Chọn nhiều nhất các số trong dãy A[i] (không nhất thiết phải liên tiếp nhau) sao tổng của chúng chia hết cho **K = 5**.

Kết quả: Chiều dài dãy con: **4**

Các phần tử được chọn là: **1 2 5 6**

Có giá trị tương ứng là: **11 6 20 8**

7.4. Bài toán dãy con có tổng chia hết cho k

Nhắc lại phép toán mod

- Giả sử **r = a mod k** và **z = b mod k**
- Ta có:

1. (a + b) mod k = (r + z) mod k
2. (-r) mod k = (-r + k) mod k
3. (r + z) mod k = v

$\rightarrow$ z = (v – r) mod k = (v – r + k) mod k

i	**1**	**2**	**3**	**4**	**5**	**6**
A[i]	11	6	7	12	20	8
A[i] = A[i] % 5	1	1	2	2	0	3
	0	0	0	3	4	4

7.4. Bài toán dãy con có tổng chia hết cho k

Tham số thể hiện kích thước bài toán

Gọi **F(i)** = chiều dài dãy con dài nhất trong miền **[1..i]** có tổng chia hết cho k.

1. Nếu dãy con dài nhất không có **A[i]** thì **F(i) = F(i-1)**
   Với F(i-1) = chiều dài dãy con dài nhất trong miền **[1..i-1]** có tổng chia hết cho k

2. Nếu dãy con dài nhất có chứa A[i]: thì **F(i) = F(i-1) + 1**
   Gọi **r = A[i] mod k**

   - Nếu **r = 0**: thì F(i-1)= chiều dài dãy con dài nhất trong miền [1..i-1] có tổng chia hết cho k

   - Nếu **r > 0**: do (r + k - r) mod k = 0 nên F(i-1)= chiều dài dãy con dài nhất trong miền [1..(i-1)] có tổng chia với k dư (k-r)

Tham số thể hiện kích thước của bài toán: kích thước miền và số dư của tổng chia với k

7.4. Bài toán dãy con có tổng chia hết cho k

Lập công thức đệ qui

Gọi **F(i, v)** = chiều dài dãy con dài nhất trong miền **[1..i]** có tổng chia với k dư là v.

1. Nếu dãy con dài nhất không có A[i] thì **F(i, v) = F(i-1, v)**
2. Nếu dãy con dài nhất có chứa A[i]:
   Gọi **r = A[i] mod k** ta có **F(i, v) = F(i-1, v - r) + 1**

   - Nếu **v – r = 0**: **F(i, v) = F(i-1, 0) + 1**

   - Nếu **v – r > 0** và **F(i-1, v - r)>0**: **F(i, v) = F(i-1, v - r) + 1**

   - Nếu **v – r < 0** và **F(i-1, v-r+k)>0**: **F(i, v) = F(i-1, v-r+k) + 1**

     Thay: **(v - r) mod k** = **(v – r + k) mod k**

Bài toán nhỏ nhất ứng với **i = 1**:

- **F(1,v)= 0** nếu **r <> v**
- **F(1,v)= 1** nếu **r = v**

7.4. Bài toán dãy con có tổng chia hết cho k

Công thức đệ qui

Gọi **r = A[i] mod k**.

- Với **i = 1**:
  - **F(1, v) = 0** nếu **r <> v**
  - **F(1, v) = 1** nếu **r = v**

- Với **i > 1**:

  - Nếu **v= r** thì: **F(i, v)= Max {F(i-1, v), F(i-1, 0) +1}**

  - Nếu **v > r** và **F(i-1, v- r) >0** thì:
    F(i, v) = Max {F(i-1, v), F(i-1, v – r) +1}

  - Nếu **v < r** và **F(i-1, v – r + k)>0** thì:
    F(i, v) = Max{F(i-1, v), F(i-1, v – r + k) + 1}

7.4. Bài toán dãy con có tổng chia hết cho k

Rút gọn công thức đệ qui

Gọi **r = A[i] mod k**.

- Với **i = 1**:
  - **F(1, v) = 0** nếu **r <> v**
  - **F(1, v) = 1** nếu **r = v**

- Với **i > 1**:

  \- Nếu **v= r** thì: **F(i, v)= Max {F(i-1, v), F(i-1, 0) +1}**

  \- Nếu **v <> r** và **F(i-1, (v-r+k) mod k) > 0** thì:
  F(i, v) = Max {F(i-1, v), F(i-1, (v – r + k) mod k) +1}

7.4. Bài toán dãy con có tổng chia hết cho k

Xây dựng bảng phương án:

❖ **Cấu trúc bảng phương án:**

Dùng mảng **F[1..n] [0..K-1]** chứa giá trị của các **F(i, v)**

❖ **Cách tính giá trị trên bảng phương án:**

Điền giá trị trên dòng **1**:

Nếu **A[1] mod k = v** thì **F[1, v] = 1** ngược lại **F[1, v] = 0**

Sử dụng công thức đệ qui và giá trị trên dòng **(i -1)** để tính dòng **i:**

- Nếu **v= r** thì: **F(i, v)= Max {F(i-1, v), F(i-1, 0) +1}**

- Nếu **v <> r** và **F(i-1, (v-r+k) mod k) > 0** thì:

F(i, v)= Max {F(i-1, v), F(i-1, (v–r+k) mod k) +1}

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: Với i = 1

- Nếu **v <> r** thì **F(1, v) = 0**
- Nếu **v = r** thì **F(1, v) = 1**

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: Với i = 2

- Nếu **v= r** thì: **F(2, v)= Max {F(1, v), F(1, 0) +1}**
- Nếu **v <> r** và **F(1, (v-r+k) mod k) > 0** thì:

F(2, v)= Max {F(1, v), F(1, (v–r+k) mod k) +1}

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	**4 | 0**	**0 | 1**	**1 | 2**	**2 | 0**	**3 | 0**

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: Với i = 3

- Nếu **v= r** thì: **F(3, v)= Max {F(2, v), F(2, 0) +1}**
- Nếu **v <> r** và **F(1, (v-r+k) mod k) > 0** thì:

F(3, v)= Max {F(2, v), F(2, (v–r+k) mod k) +1}

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	4 | 0	0 | 1	1 | 2	2 | 0	3 | 0
3	2	**3 | 0**	**4 | 1**	**0 | 2**	**1 | 2**	**2 | 3**

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: **Với i = 4**

- Nếu **v= r** thì: **F(4, v)= Max {F(3, v), F(3, 0) +1}**
- Nếu **v <> r** và **F(1, (v-r+k) mod k) > 0** thì:

  F(4, v)= Max {F(3, v), F(3, (v–r+k) mod k) +1}

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	4 | 0	0 | 1	1 | 2	2 | 0	3 | 0
3	2	3 | 0	4 | 1	0 | 2	1 | 2	2 | 3
4	2	**3 | 3**	**4 | 3**	**0 | 2**	**1 | 2**	**2 | 3**

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: Với i = 5

- Nếu **v= r** thì: **F(5, v)= Max {F(4, v), F(4, 0) +1}**
- Nếu **v <> r** và **F(1, (v-r+k) mod k) > 0** thì:

F(5, v)= Max {F(4, v), F(4, (v–r+k) mod k) +1}

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	4 | 0	0 | 1	1 | 2	2 | 0	3 | 0
3	2	3 | 0	4 | 1	0 | 2	1 | 2	2 | 3
4	2	3 | 3	4 | 3	0 | 2	1 | 2	2 | 3
5	0	**0 | 4**	**1 | 4**	**2 | 3**	**3 | 3**	**4 | 4**

7.4. Bài toán dãy con có tổng chia hết cho k

Ví dụ bảng phương án: Với i = 6

- Nếu **v= r** thì: **F(6, v)= Max {F(5, v), F(5, 0) +1}**
- Nếu **v <> r** và **F(1, (v-r+k) mod k) > 0** thì:

  F(6, v)= Max {F(5, v), F(5, (v–r+k) mod k) +1}

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	4 | 0	0 | 1	1 | 2	2 | 0	3 | 0
3	2	3 | 0	4 | 1	0 | 2	1 | 2	2 | 3
4	2	3 | 3	4 | 3	0 | 2	1 | 2	2 | 3
5	0	0 | 4	1 | 4	2 | 3	3 | 3	4 | 4
6	3	**2 | 4**	**3 | 4**	**4 | 5**	**1 | 5**	**2 | 6**

7.4. Bài toán dãy con có tổng chia hết cho k

Thuật toán tạo bảng phương án

```
void TaoBangPhuongAn(F [1...n] [0...k-1])
{  for (v=0; v <= M; v++)
         F[1, v] = (A[i]%k == v) ? 1 : 0;
   for (i = 2; i <= n; i++)
      for (v=0; v <= k-1; v++)
      {  r = A[i] % k; F[i, v] = F[i-1, v];
         if (v == r && F[i, v] <= F[i-1, 0 ])
            F[i, v] = F[ i-1, 0] ] + 1;
         else
         if   (F[i-1, (v - r + k)%k] >0 && F[i, v] <=
            F[i-1, (v - r + k)%k])
         F[i, v] = F[ i -1, (v - r + k)%k] + 1;
      }
}
```

7.4. Bài toán dãy con có tổng chia hết cho k

Truy vết:

Bắt đầu từ ô **F[n, 0]** trên dòng n ta dò ngược về dòng 0 theo nguyên tắc:

- Nếu **F[i–1, (v-r+k)% k] > 0** và **F[i, v] > F[i–1, (v-r+k)% k]:**
  - **A[i]** được chọn
  - Truy tiếp ô **F[i -1, (v-r+k)% k].**
- Ngược lại thì **A[i]** không được chọn, truy tiếp ô **F[i -1, v]**.

i	r=A[i] v	0	1	2	3	4
1	1	0	1	0	0	0
2	1	4 | 0	0 | 1	1 | 2	2 | 0	3 | 0
3	2	3 | 0	4 | 1	0 | 2	1 | 2	2 | 3
4	2	3 | 3	4 | 3	0 | 2	1 | 2	2 | 3
5	0	0 | 4	1 | 4	2 | 3	3 | 3	4 | 4
6	3	**2 | 4**	**3 | 4**	**4 | 5**	**1 | 5**	**2 | 6**

7.4. Bài toán dãy con có tổng chia hết cho k

Thuật toán truy vết

```
void TruyVet(F [i, v])
{    Bắt đầu từ ô F[n, 0] trên dòng n: i =
  n; v = 0;
  for (; i > 0; i --)
    if(F[i-1, (v-r+k)% k] >0 && F[i, v]>
    F[i-1, (v-r+k)% k])
     { <A[i] được chọn>;
       v = (v - r + k)% k;
     }
}
```

Tổng kết

Bài toán ba lô 2

- Tham số thể hiện kích thước bài toán
- Lập công thức đệ quy
- Xây dựng bảng phương án lựa chọn các gói hàng $\rightarrow$ p/ án tối ưu
- Truy lại các gói hàng cùng s/l đã chọn

Bài toán dãy con nhỏ nhất chia hết cho k

- Tham số thể hiện kích thước bài toán
- Lập công thức đệ quy
- Xây dựng bảng phương án lựa chọn các số trong dãy $\rightarrow$ p/ án tối ưu
- Truy vết tìm lại các số đã chọn

Bài tập

1. Cho một balo có khối lượng M= 11.

   Có 4 đồ vật lần lượt như sau: khối lượng A[i]={4,2,4,3} với giá trị tương ứng C[i]={1,5,2,7}.

   Hãy chọn các đồ vật để cho vào balo sao cho các đồ vật được chọn có giá trị lớn nhất và khối lượng không vượt quá khối lượng của balo. Biết rằng mỗi đồ vật có thể được chọn nhiều lần.

Bài tập

2. Cho một dãy A gồm 8 số nguyên như sau

13	7	9	8	24	6	12	20

và một số nguyên dương k=8.

Hãy tìm một dãy con (không nhất thiết phải liên tiếp nhau) dài nhất có tổng các số chia hết cho số k.

Tài liệu tham khảo

[1]. Giáo trình Cấu trúc dữ liệu và giải thuật – Lê Văn Vinh, NXB Đại học quốc gia TP HCM, 2013

[2]. Cấu trúc dữ liệu & thuật toán, Đỗ Xuân Lôi, NXB Đại học quốc gia Hà Nội, 2010.

[3]. Trần Thông Quế, *Cấu trúc dữ liệu và thuật toán (phân tích và cài đặt trên C/C++)*, NXB Thông tin và truyền thông, 2018

[4]. Robert Sedgewick, *Cẩm nang thuật toán*, NXB Khoa học kỹ thuật, 2004 .

[5]. PGS.TS Hoàng Nghĩa Tý, *Cấu trúc dữ liệu và thuật toán*, NXB xây dựng, 2014