Tuần 1

Môn: **Lý thuyết đồ thị và Tổ hợp**

Đề bài a: Chứng minh rằng với mọi $n \in \mathbb{N}^+$, ta có:

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

Chứng minh: Ta sẽ sử dụng phương pháp quy nạp toán học.

Bước 1: Cơ sở quy nạp.
Xét $n = 1$:

$$\sum_{i=1}^{1} i = 1 \quad \text{và} \quad \frac{1(1+1)}{2} = \frac{2}{2} = 1$$

Mệnh đề đúng với $n = 1$.

Bước 2: Giả thiết quy nạp.
Giả sử mệnh đề đúng với $n = k$, tức là:

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

Bước 3: Bước quy nạp.
Ta cần chứng minh mệnh đề đúng với $n = k + 1$:

$$\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

Ta có:

$$\sum_{i=1}^{k+1} i = \left(\sum_{i=1}^{k} i\right) + (k+1) = \frac{k(k+1)}{2} + (k+1)$$
$$= \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$$

Mệnh đề đúng với $n = k + 1$.

Đề bài b: Chứng minh rằng với mọi $n \in \mathbb{N}^+$, ta có:

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Chứng minh: Ta sử dụng phương pháp quy nạp toán học.

Bước 1: Cơ sở quy nạp.
Với $n = 1$:

$$\sum_{i=1}^{1} i^2 = 1^2 = 1 \quad \text{và} \quad \frac{1(1+1)(2\cdot 1+1)}{6} = \frac{1\cdot 2\cdot 3}{6} = 1$$

Mệnh đề đúng với $n = 1$.

Bước 2: Giả thiết quy nạp.
Giả sử mệnh đề đúng với $n = k$, tức là:

$$\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$$

Bước 3: Bước quy nạp.
Xét $n = k + 1$, ta cần chứng minh:

$$\sum_{i=1}^{k+1} i^2 = \frac{(k+1)(k+2)(2k+3)}{6}$$

Ta có:

$$\sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

$$= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} = \frac{(k+1)\left[k(2k+1) + 6(k+1)\right]}{6}$$

$$= \frac{(k+1)(2k^2 + 7k + 6)}{6} = \frac{(k+1)(k+2)(2k+3)}{6}$$

Mệnh đề đúng với $n = k + 1$.

Kết luận: Theo nguyên lý quy nạp toán học, mệnh đề đúng với mọi $n \in \mathbb{N}^+$.

Đề bài c: Chứng minh rằng với mọi $n \in \mathbb{N}^+$, ta có:

$$\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

Chứng minh: Ta sử dụng phương pháp quy nạp.

Bước 1: Cơ sở quy nạp.
Với $n = 1$:

$$\sum_{i=1}^{1} i^3 = 1^3 = 1, \quad \frac{1^2(1+1)^2}{4} = \frac{4}{4} = 1$$

Mệnh đề đúng với $n = 1$.

Bước 2: Giả thiết quy nạp.
Giả sử mệnh đề đúng với $n = k$:

$$\sum_{i=1}^{k} i^3 = \frac{k^2(k+1)^2}{4}$$

Bước 3: Bước quy nạp.

Ta chứng minh mệnh đề đúng với $n = k + 1$:

$$\sum_{i=1}^{k+1} i^3 = \sum_{i=1}^{k} i^3 + (k+1)^3 = \frac{k^2(k+1)^2}{4} + (k+1)^3$$

$$= (k+1)^2 \left(\frac{k^2}{4} + (k+1) \right) = (k+1)^2 \cdot \frac{k^2+4k+4}{4} = \frac{(k+1)^2(k+2)^2}{4}$$

Mệnh đề đúng với $n = k + 1$.

Đề bài e: Chứng minh rằng với mọi $n \in \mathbb{N}^+$, ta có:

$$\sum_{i=1}^{n} (2i-1) = n^2$$

Chứng minh: Ta sử dụng phương pháp quy nạp.

Bước 1: Cơ sở quy nạp.
Với $n = 1$:

$$\sum_{i=1}^{1} (2i-1) = 1, \quad 1^2 = 1 \Rightarrow \text{Mệnh đề đúng với } n = 1.$$

Bước 2: Giả thiết quy nạp.
Giả sử mệnh đề đúng với $n = k$, tức là:

$$\sum_{i=1}^{k} (2i-1) = k^2$$

Bước 3: Bước quy nạp.
Ta cần chứng minh:

$$\sum_{i=1}^{k+1} (2i-1) = (k+1)^2$$

Ta có:

$$\sum_{i=1}^{k+1} (2i-1) = \left(\sum_{i=1}^{k} (2i-1) \right) + (2(k+1)-1) = k^2 + (2k+1) = (k+1)^2$$

Vậy mệnh đề đúng với $n = k + 1$.

Bài g. Tính hai tổng sau:

$$\sum_{i=1}^{n} (2i-1)^2 \quad \text{và} \quad \sum_{i=1}^{n} (2i)^3$$

1. Tổng của các số lẻ bình phương:

$$\sum_{i=1}^{n}(2i-1)^2$$

Khai triển:

$$(2i-1)^2 = 4i^2 - 4i + 1 \Rightarrow \sum_{i=1}^{n}(2i-1)^2 = 4\sum_{i=1}^{n} i^2 - 4\sum_{i=1}^{n} i + \sum_{i=1}^{n} 1$$

Áp dụng công thức:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^{n} 1 = n$$

Ta có:

$$= 4 \cdot \frac{n(n+1)(2n+1)}{6} - 4 \cdot \frac{n(n+1)}{2} + n = \frac{2n(n+1)(2n+1)}{3} - 2n(n+1) + n$$

$$= \frac{2n(n+1)(2n+1) - 6n(n+1) + 3n}{3} = \frac{n(4n^2-1)}{3}$$

Vậy:

$$\sum_{i=1}^{n}(2i-1)^2 = \frac{n(4n^2-1)}{3}$$

2. Tổng lập phương của các số chẵn:

$$\sum_{i=1}^{n}(2i)^3 = 8\sum_{i=1}^{n} i^3 = 8 \cdot \left(\frac{n(n+1)}{2}\right)^2 = 2n^2(n+1)^2$$

Vậy:

$$\sum_{i=1}^{n}(2i)^3 = 2n^2(n+1)^2$$

Đề bài h: Tính tổng:

$$\sum_{i=1}^{n}(2i-1)^3$$

Lời giải:
Tổng trên là tổng lập phương của n số lẻ đầu tiên:

$$1^3 + 3^3 + 5^3 + \cdots + (2n-1)^3$$

Ta có công thức:

$$\sum_{i=1}^{n}(2i-1)^3 = n^2(2n^2-1)$$

Chứng minh:
Khai triển:

$$(2i-1)^3 = 8i^3 - 12i^2 + 6i - 1 \Rightarrow \sum_{i=1}^{n}(2i-1)^3 = 8\sum i^3 - 12\sum i^2 + 6\sum i - \sum 1$$

Thay công thức:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2, \quad \sum 1 = n$$

Từ đó:

$$\sum_{i=1}^{n}(2i-1)^3 = 8\cdot\left(\frac{n(n+1)}{2}\right)^2 - 12\cdot\frac{n(n+1)(2n+1)}{6} + 6\cdot\frac{n(n+1)}{2} - n$$

Rút gọn biểu thức trên, ta được:

$$\sum_{i=1}^{n}(2i-1)^3 = n^2(2n^2-1)$$

BÀI 1: In Pascal Triangle & Khai triển Newton

1. Tính tổ hợp và xây dựng Tam giác Pascal:

Dưới đây là đoạn mã C++ sử dụng đệ quy để tính tổ hợp $C(n,k)$ và in ra tam giác Pascal đến dòng thứ n:

```
#include <iostream>
using namespace std;

int C(int n, int k) {
    if (k == 0 || k == n) return 1;
    return C(n - 1, k - 1) + C(n - 1, k);
}

void printPascal(int n) {
    for (int i = 0; i <= n; ++i) {
        for (int k = 0; k <= i; ++k)
            cout << C(i, k) << " ";
        cout << endl;
    }
}

void binomialExpansion(int n) {
    cout << "(a + b)^" << n << " = ";
    for (int k = 0; k <= n; ++k) {
```

```
        cout << C(n, k) << "*a^" << n - k << "*b^" << k;
        if (k < n) cout << " + ";
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Nhap n: ";
    cin >> n;
    cout << "Tam giac Pascal:\n";
    printPascal(n);
    binomialExpansion(n);
    return 0;
}
```

Giải thích chi tiết chương trình

1. Hàm tính tổ hợp $C(n, k)$:

```
int C(int n, int k) {
    if (k == 0 || k == n) return 1;
    return C(n - 1, k - 1) + C(n - 1, k);
}
```

- Đây là hàm đệ quy tính tổ hợp theo định nghĩa:

$$C(n,k) = \begin{cases} 1 & \text{nếu } k = 0 \text{ hoặc } k = n \\ C(n-1,k-1) + C(n-1,k) & \text{ngược lại} \end{cases}$$

- Khi gọi `C(5, 2)`, chương trình sẽ tự động chia nhỏ thành các lời gọi con cho đến khi gặp các trường hợp cơ sở (`k == 0` hoặc `k == n`).

2. Hàm in tam giác Pascal:

```
void printPascal(int n) {
    for (int i = 0; i <= n; ++i) {
        for (int k = 0; k <= i; ++k)
            cout << C(i, k) << " ";
        cout << endl;
    }
}
```

- Hàm này in ra từng dòng của tam giác Pascal từ dòng 0 đến n.
- Với mỗi dòng i, ta in các giá trị $C(i, 0), C(i, 1), \ldots, C(i, i)$.
- Ví dụ nếu $n = 3$, đầu ra sẽ là:

```
1
1 1
1 2 1
1 3 3 1
```

3. Hàm khai triển nhị thức Newton $(a+b)^n$:

```
void binomialExpansion(int n) {
    cout << "(a + b)^" << n << " = ";
    for (int k = 0; k <= n; ++k) {
        cout << C(n, k) << "*a^" << n - k << "*b^" << k;
        if (k < n) cout << " + ";
    }
    cout << endl;
}
```

- Dựa theo công thức khai triển Newton:

$$(a+b)^n = \sum_{k=0}^{n} C(n,k) \cdot a^{n-k} \cdot b^k$$

- Hàm duyệt từ $k = 0$ đến n và in từng hạng tử theo đúng định dạng $C(n,k)a^{n-k}b^k$.
- Nếu $n = 2$ thì đầu ra sẽ là:

$$(a+b)^2 = 1 * a^2 * b^0 + 2 * a^1 * b^1 + 1 * a^0 * b^2$$

4. Hàm `main()`:

```
int main() {
    int n;
    cout << "Nhap n: ";
    cin >> n;
    cout << "Tam giac Pascal:\n";
    printPascal(n);
    binomialExpansion(n);
    return 0;
}
```

- Chương trình yêu cầu người dùng nhập vào một số nguyên n.
- Sau đó in ra tam giác Pascal và khai triển nhị thức Newton với số mũ n.

Bài toán 2: Tổ hợp và kiểm tra overflow

Yêu cầu: Viết chương trình bằng C/C++ để:

- Tính các giá trị: $P_n = n!$, $A_n^k = \frac{n!}{(n-k)!}$, $C_n^k = \frac{n!}{k!(n-k)!}$, và số Catalan thứ n:

$$C_n = \frac{(2n)!}{(n+1)! \cdot n!}$$

- Xác định giá trị nhỏ nhất của n tại đó xảy ra **overflow**.

Chương trình C++:

```
#include <iostream>
#include <climits>
using namespace std;

unsigned long long factorial(int n) {
    unsigned long long res = 1;
    for (int i = 2; i <= n; ++i) {
        if (res > ULLONG_MAX / i) {
            cout << "Overflow tại n = " << i << " khi tính giai thừa.\n";
            return 0;
        }
        res *= i;
    }
    return res;
}

unsigned long long permutation(int n, int k) {
    unsigned long long a = factorial(n);
    unsigned long long b = factorial(n - k);
    return (b == 0) ? 0 : a / b;
}

unsigned long long combination(int n, int k) {
    unsigned long long a = factorial(n);
    unsigned long long b = factorial(k);
    unsigned long long c = factorial(n - k);
    return (b == 0 || c == 0) ? 0 : a / (b * c);
}

unsigned long long catalan(int n) {
    unsigned long long a = factorial(2 * n);
    unsigned long long b = factorial(n + 1);
    unsigned long long c = factorial(n);
    return (b == 0 || c == 0) ? 0 : a / (b * c);
}
```

```
int main() {
    int n, k;
    cout << "Nhap n (n > 0): ";
    cin >> n;
    cout << "Nhap k (0 <= k <= n): ";
    cin >> k;

    cout << "\n=== Ket qua ===\n";

    unsigned long long fn = factorial(n);
    if (fn != 0) cout << "Pn = " << fn << endl;

    unsigned long long Ank = permutation(n, k);
    if (Ank != 0) cout << "A^k_n = " << Ank << endl;

    unsigned long long Cnk = combination(n, k);
    if (Cnk != 0) cout << "C^k_n = " << Cnk << endl;

    unsigned long long Catn = catalan(n);
    if (Catn != 0) cout << "Catalan(n) = " << Catn << endl;

    return 0;
}
```

Giải thích chi tiết chương trình:

- **Hàm** `factorial(int n)`: Tính $n!$ và kiểm tra nếu tại bất kỳ bước nào $res >$ `ULLONG_MAX`$/i$, tức là phép nhân tiếp theo sẽ gây tràn số. Khi đó chương trình in ra cảnh báo overflow và trả về 0.
- **Hàm** `permutation(n, k)`: Tính hoán vị $A_n^k = \frac{n!}{(n-k)!}$ bằng cách gọi `factorial()`.
- **Hàm** `combination(n, k)`: Tính tổ hợp $C_n^k = \frac{n!}{k!(n-k)!}$. Nếu mẫu số bị overflow thì trả về 0 để tránh sai kết quả.
- **Hàm** `catalan(n)`: Tính số Catalan thứ n theo công thức:

$$C_n = \frac{(2n)!}{(n+1)! \cdot n!}$$

- **Hàm** `main()`: Cho người dùng nhập n và k, sau đó hiển thị:
  - $P_n = n!$
  - A_n^k
  - C_n^k
  - Số Catalan thứ n
- Nếu bất kỳ phép toán nào gây tràn số thì chương trình sẽ in ra thông báo `Overflow tại n = ...`.