BÀI THỰC HÀNH TOÁN TỔ HỢP

PHẦN A. TÍNH TOÁN SỐ HỌC

Thực hiện các bài tập sau đây bằng 3 cách: đề xuất thuật toán và viết chương trình *C/C++* (hay *C#, Java*), gọi hàm có sẵn của *MathLab*/*Julia*, viết chương trình trong *MathLab*/*Julia* (với thuật toán đã đề xuất). Chạy thử với dữ liệu nhập là số khá lớn. Sinh viên chỉ cần chọn 2 bài mình thích.

1. Với n nguyên dương, hãy phân tích $n!$ ra thừa số nguyên tố.
2. Với n nguyên dương, k nguyên và $0 \le k \le n$, hãy phân tích $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ ra thừa số nguyên tố.

Ý tưởng chính của hai bài trên là nếu tính toán thông qua việc tính $n!$ thì sẽ rất chậm và có nguy cơ bị tràn số, bởi vì n chỉ cần vài ngàn thì $n!$ đã rất lớn. Trong khi đó $n!$ chỉ có các ước số nguyên tố khá nhỏ (Tại sao như vậy?).

3. Tính (chính xác) và in ra chữ số thứ n sau dấy phẩy trong biểu diễn bát phân (hệ đếm cơ số 8) của số π.

   Khi n lớn (giả sử vài trăm tỷ) thì các thuật toán thông thường rất chậm (do phải duyệt tuần tự qua các chữ số phía trước của chữ số cần tìm) và không chính xác (nếu dựa vào tính toán số thực *floating point*). Vì vậy để có kết quả chính xác và tốc độ cao ta có thể dựa vào công thức nổi tiếng, chỉ mới vừa được công bố trong vài năm gần đây:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left[\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right]$$

4. Số e được tính bằng công thức: $e = \sum_{k=0}^{\infty} \frac{1}{k!}$. Hãy dựa vào công thức này (hoặc là tìm một công thức tương tự, chẳng hạn như $e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ hay một công thức khác có thể tính toán nhanh hơn) để tính chính xác chữ số thứ k sau dấu phẩy của số e trong một hệ đếm nào đó.

PHẦN A. LÝ THUYẾT ĐỒ THỊ

Sinh viên sẽ được hướng dẫn cài đặt 02 thuật toán lý thuyết đồ thị bằng *C/C++* (hay *C#, Java*) và *MathLab*/*Julia*,