TEN 2011. Esercizi 3. Fattorizzazione per tentativi, Pollard ρ, Pollard $p-1$.

Notazione: Indichiamo con $\log n$ il logaritmo di n in base 2 e con $\ln n$ il logaritmo naturale di n, in base e.
Attenzione: i links ai file vanno ribattuti completamente (col copia-incolla non funzionano).

1. Il metodo di fattorizzazione per tentativi (*trial division*) consiste nel dividere un numero intero $n \geq 0$ per tutti i primi $p \leq \sqrt{n}$, in ordine crescente fino a che non si trova un fattore.
   (a) Sia n un intero. Quanti sono approssimativamente i primi $p \leq \sqrt{n}$?
   (b) Qual è la complessità di una divisione di n per un primo $p \leq \sqrt{n}$?
   (c) Verificare che la complessità del metodo di fattorizzazione per tentativi si può stimare come

$$\mathcal{O}(p \log n),$$

   dove n è l'intero da fattorizzare e p è il più piccolo fattore di n (usare (a) e (b)).

2. Sia n un intero da fattorizzare. Stimare il numero di operazioni necessarie a fattorizzare un intero n col metodo di fattorizzazione per tentativi ogni volta che n diventa dieci volte più grande (ossia ha una cifra decimale in più): osservare la differenza fra il caso in cui il fattore più piccolo p ha una cifra decimale in più e il caso in cui il quoziente $\frac{n}{p}$ ha una cifra decimale in più (mentre p resta fisso).
   Esempio1: `http://www.mat.uniroma2.it/~geo2/naif-exper1.txt`
   Esempio2: `http://www.mat.uniroma2.it/~geo2/naif-exper2.txt`

3. Siano dati i numeri

$$n1 = nextprime(10^5) * nextprime(10^{10}), \quad n2 = nextprime(10^6) * nextprime(10^{10}).$$

   Fattorizzare $n1$ ed $n2$ con "trial division". Sulla base del tempo impiegato, esibire degli interi la cui fattorizzazione con tale algoritmo richiede un minuto, un'ora, un giorno, un mese, un anno,....., un secolo.

4. Siano dati i numeri

$$n1 = nextprime(10^5) * nextprime(10^{10}), \quad n2 = nextprime(10^6) * nextprime(10^{10}),$$
$$n3 = nextprime(10^7) * nextprime(10^{10}), \quad n3 = nextprime(10^8) * nextprime(10^{10}),$$
$$n5 = nextprime(10^{10}) * nextprime(10^{50}), \quad n6 = nextprime(10^{10}) * nextprime(10^{100}),$$
$$n7 = nextprime(10^{10}) * nextprime(10^{200}), \quad n8 = nextprime(10^{10}) * nextprime(10^{400}).$$

   (a) Fattorizzarli col metodo ρ di Pollard.
   (b) Quante iterate dobbiamo aspettarci di fare nei singoli casi?
   (c) Come varia il numero di operazioni necessarie a fattorizzare $n5$, $n6$, $n7$, $n8$?

5. Implementare l'algoritmo ρ di Pollard e usarlo per
   (a) fattorizzare i numeri di Mersenne M_n per $1 \leq n \leq 60$;
   (b) fattorizzare i numeri di Fermat F_n, per $1 \leq n \leq 8$.

6. Sia n un numero intero composto di 200 cifre. Dopo 10000 iterazioni l'algoritmo di Pollard ρ non ha trovato nessun fattore. Cosa possiamo concludere?

7. Sia n un numero intero composto di 200 cifre, e sia $B = 10000$. Dopo quante iterazioni dell'algoritmo di Pollard ρ possiamo verosimilmente che escludere che n sia B-smooth ?

8. Sia n un intero da fattorizzare. Stimare il numero di operazioni necessarie a fattorizzare un intero n col metodo ρ di Pollard ogni volta che n diventa dieci volte più grande: osservare la differenza fra il caso in cui il fattore più piccolo p ha una cifra decimale in piu' e il caso in cui il quoziente $\frac{n}{p}$ ha una cifra decimale in più (mentre p resta fisso).
   Esempio1: `http://www.mat.uniroma2.it/~geo2/pollard-exper1.txt`
   N.B.: Nel caso del metodo ρ di Pollard la stima è probabilistica: se ripetiamo il calcolo con gli stessi dati, impieghiamo sempre lo stesso tempo?

9. Sia n un intero da fattorizzare. Stimare la complessità (probabilistica) del metodo di fattorizzazione che consiste nel calcolare il massimo comun divisore fra n e un numero a caso $1 \leq m \leq n$.

10. Esercizio col metodo $p-1$ di Pollard (Usare PARI/GP e l'applet per la fattorizzazione on-line di Alpertron).
    (a) Fattorizzare i seguenti 3 numeri col metodo $p-1$ di Pollard. Usare ad esempio:
    `http://www.mat.uniroma2.it/~geo2/pminus.txt`

$n = 39080295191118915018134958938415108346749622881999563438557941763777 38$
$37879970068136035919305517302338111572218251 71$

$n = 54776770731927734524510635129304900304647682454932386722232624524088 99$
$11619631465164124733005068230074356253946038850399554060231365982075 03$

$n = 10000000000000000038479003457230000013485823303075317000000004662354690 63$

    (b) Controllare la primalità dei fattori trovati al punto (a) ed eventualmente fattorizzarli.
    (c) Ripetere l'algoritmo $p-1$ di Pollard e verificare che nei casi in cui ha successo, generalmente spezza il numero come $n = m * q$ dove m è il prodotto di tutti i fattori primi p, per cui $p-1$ è B-smooth (in questo caso provare con $B = 1500$).
    (d) Sperimentare con l'algoritmo $p-1$ di Pollard, scegliendo diversi valori di B.

11. (*Pollard* $p-1$). Sia $M = 2^3 \cdot 3^2 \cdot 5 \cdot 7$.
    (a) Sia $n = 95431706263$. Scegliere $\overline{a} \in \mathbb{Z}_n^*$ a caso. Calcolare $\overline{b} = \overline{a}^M \bmod n$. Calcolare il divisore $d = \mathrm{mcd}(b-1, n)$ di n ed il cofattore n/d.
    (b) Sia $n = 57841557763361$. Scegliere $\overline{a} \in \mathbb{Z}_n^*$ a caso. Calcolare $\overline{b} = \overline{a}^M \bmod n$. Calcolare il divisore $d = \mathrm{mcd}(b-1, n)$ di n ed il cofattore n/d.
    (c) Come mai l'algoritmo trova queste due fattorizzazioni?