Corso Tecniche di Programmazione Avanzata
Esame del 8 luglio 2019

Esercizio A

Problema

Una espressione aritmetica è composta dai soli caratteri

$$1 \qquad + \qquad * \qquad (\qquad)$$

ogni intero positivo può essere scritto in più modi come una espressione che usa i caratteri precedenti. Ad esempio il numero 14 può essere scritto come:

- $1+1+1+1+1+1+1+1+1+1+1+1+1+1$
- $((1+1)*(1+1+1+1+1))+((1+1)*(1+1))$
- $(1+1)*(1+1+1+1+1+1+1)$
- $(1+1)*(((1+1+1)*(1+1))+1)$

Calcolare il numero MINIMO di 1 che devono essere usati per calcolare un numero intero positivo assegnato. Le parentesi i "$+$" e i "$*$" non si contano. Per esempio, quando $n = 14$, l'algoritmo deve restituire 8. Sia $\Phi(n)$ la funzione che restituisce un minimo numero di 1 allora, l'algoritmo deve restituire $\Phi(n)$. Completare quindi la funzione

```
unsigned
phi( unsigned n ) {
  // da completare
}
```

Suggerimento

Sia $\Phi(n)$ la funzione che restituisce un minimo numero di 1 allora, se $n = 1$ l'algoritmo deve restituire $\Phi(1) = 1$. Per $n > 0$ possiamo considerare tutti i possibili modi di scrivere n come prodotto di 2 numeri $pq = n$ o come somma $p + q$ dove p e q sono interi positivi. Quindi $\Phi(n)$ soddisfa la ricorsione

$$\Phi(n) = \begin{cases} 1 & \text{se } n = 1 \\ \min\left\{ \min_{pq=n} \Phi(p) + \Phi(q), \quad \min_{p+q=n} \Phi(p) + \Phi(q) \right\} & \text{altrimenti} \end{cases}$$

Per ottimizzare la ricorsione usare la memoizzazione (programmazione dinamica). Altrimenti i tempi diventano geologici. Per velocizzare ulteriormente non usare una `map` ma un `vector`. Osservare inoltre che quando partendo da $p = 1$ quando $p > q$ rifate gli stessi conti.

Punteggio

Ci sono 8 test, ogni test da un punteggio 1.0 se il problema è risolto 0 altrimenti.

Esempio

Alcuni numeri per esercitarsi e debuggare il codice.

n	3	5	7	12	19	24	31	44
$\Phi(n)$	3	5	6	7	9	9	11	12