3.4 ⋆ Numeri Semiperfetti

Probabilmente tutti conoscete i numeri perfetti: un intero positivo n si dice *perfetto* se è uguale alla somma di tutti i suoi divisori propri. Ad esempio, $28 = 1 + 2 + 4 + 7 + 14$ è perfetto. Potreste, per fare amicizia coi perfetti, dimostrare che se $2^n - 1$ è primo, allora $2^{n-1} \times (2^n - 1)$ è perfetto[8]. Questo era già noto ad Euclide, mentre Eulero dimostrò che tutti i perfetti pari sono in questa forma. Non si conoscono, al momento, perfetti dispari.

Classifichiamo ora tutti i numeri: n si dice *abbondante* se la somma di tutti i suoi divisori propri è maggiore di n. Ad esempio $24 < 1 + 2 + 3 + 4 + 6 + 8 + 12 = 36$ è abbondante. n si dice *difettivo* se la somma di tutti i suoi divisori propri è minore di n. Ad esempio $27 > 1 + 3 + 9 = 13$ è difettivo. Tra i numeri abbondanti, alcuni sono detti *semiperfetti*, se sono uguali alla somma di di *alcuni* dei loro divisori propri (non necessariamente tutti). Ad esempio il 24 può essere scritto come $4 + 8 + 12$ (in realtà anche in altri modi).

È facile dimostrare che tutti i multipli di un perfetto o di un semiperfetto sono anch'essi semiperfetti, ma (ciò rende il seguito interessante) ci sono numeri *semiperfetti primari*, che non sono cioè multipli di perfetti o di altri semiperfetti (ad esempio il 40). E ci sono anche semiperfetti dispari ragionevolmente piccoli (il primo è il 945).

Poniamoci il problema di scrivere un programma che determina se un numero è semiperfetto. Indichiamo con $D^n = \{d_1, \ldots, d_m\}$ l'insieme degli m divisori propri di n e supponiamo, per semplicità di scegliere gli indici dimodoché $d_1 < d_2 < \ldots < d_m$ ($d_m < n$). Chiamiamo $\sharp D$ la sommatoria degli elementi di D. Formalmente, preso un naturale n, cerchiamo se esiste un insieme $D \subseteq D^n$, tale che $\sharp D = n$.

Proviamo a vedere se riusciamo a costruire incrementalmente (se esiste) un tale D, inserendo i divisori di n in ordine crescente. Indichiamo ora (per $1 \leq k \leq m$), con $D^{n,k}$ l'insieme dei primi k divisori di n. Supponiamo di aver selezionato ad un certo punto, un insieme $D \subseteq D^{n,k}$. Ora è chiaro che se $\sharp D > n$ non ci sono sovrainsiemi di D che soddisfano la proprietà cercata. Se $\sharp D = n$, allora D è la prova che n è semiperfetto. Infine se $\sharp D < n$, devo ancora distinguere due casi: se ho già finito i divisori ($d_k = d_m$) non posso estendere D con elementi più grandi di quelli che già vi sono, come ci ervamo proposti. Infine, potrebbe esistere D' tale che $\sharp D' = n$ e $D \subset D' \subseteq D^{n,k}$. Tale D' potrebbe contenere d_{k+1} oppure no.

Osserviamo che in tutti questi ragionamenti (visto che abbiamo fissato un ordine nella nostra ricerca) non è rilevante sapere *quali siano gli elementi* dell'insieme D, quanto semplicemente il valore di $\sharp D$ e l'insieme $D^{n,k}$ esplorato finora. $D^{n,k}$ peraltro,

[8] I primi nella forma $2^n - 1$ sono detti *primi di Mersenne*, dal nome del monaco francese che li ha studiati. Se ne conoscono circa una quarantina, e spesso il record del numero primo più grande trovato è detenuto da un primo di Mersenne. Trovare primi molto grandi, non è solo un sublime gioco matematico/informatico, ma è molto rilevante in Crittografia.

Tornando alle cose belle e inutili, credo che tutti i perfetti noti, siano perfetti divisibili per un primo di Mersenne. Va detto infine che non è noto se i primi di Mersenne (e i perfetti) siano infiniti.

```
int semiperfettoAux(int d, int s, int n){
/* PREC: d<n & exists d_1, ..., d_k < d divisori di n, s=d_1+...+d_k */
    int j;
    for (j=d; n%j!=0; j++); /* trova prossimo divisore. Alla peggio e' n */
    if (j==n) return 0; /* se j e' n non ci sono altri divisori propri */
    if (j+s > n) return 0; /* la strada della semiperfezione e' perduta */
    if (j+s == n) {printf("%d ",j); return 1;} /* ho trovato una soluzione */
    if (semiperfettoAux(j+1, s+j, n)) return 1; /* provo a inserire j in D */
    return (semiperfettoAux(j+1, s, n)); /* provo senza inserire j in D */
    }
```

Figura 6: verifica se un numero è semiperfetto

è completamente caratterizzato dal numero d_k. Fatte queste osservazioni, possiamo immaginare di scrivere una procedura:

`int semiperfettoAux(int d_k, int s, int n)`

tale che $d_k < n$ rappresenta l'insieme di divisori $D^{n,k}$ esplorato finora, $s = \sharp D$ rappresenta l'insieme selezionato finora, ed n rappresenta il numero di cui stiamo valutando la semiperfezione. La chiamata a una tale procedura ha senso se $s < n$ e $\exists D \subseteq D^{n,k}.\sharp D = s$ e $d_k < n$. A questo punto non resta che cercare il prossimo divisore d_{k+1} di n e:

- se $d_{k+1} + s = n$ allora terminare con successo;
- se $d_{k+1} = n$ allora terminare con insuccesso perché non ci sono più divisori propri;
- se $d_{k+1} + s < n$ continuare la ricerca, attivando due chiamate ricorsive, una inserendo d_{k+1} nel potenziale insieme di divisori che certifica la semiperfezione, e una ignorandolo.

A questo punto, la funzione `semiperfettoAux` si dovrebbe scrivere con una certa facilità, come in Fig. 6.

La funzione `int semiperfetto(int n)` si limiterà a impostare la prima chiamata, assicurandosi di soddisfare le precondizioni sopra discusse. Chiaramente, all'inizio l'insieme selezionato è $D = \varnothing \subseteq D^{n,1}$: di conseguenza, il primo divisore da considerare è l'1 e $\sharp D = 0$. Quindi, ecco il codice:

```
int semiperfetto(int n){
    return semiperfettoAux(1,0,n);
}
```

Provate a visualizzare la sequenza di chiamate ricorsive, magari per un numero non semiperfetto (per esempio il 27) e per uno semiperfetto. Provate poi a modificare la funzione `semiperfettoAux` in modo che venga stampato un insieme di divisori di somma n (è sufficiente aggiungere delle stampe nei punti giusti). Gli audaci, provino a modificare la funzione `semiperfettoAux` dimodoché trovi *tutti* gli insiemi di divisori di n di somma n.