

15.7 Ricerca di un sottoalbero

Supponiamo di dover affrontare un ulteriore problema: dato l'albero allocato col procedimento del paragrafo precedente, ricercare il valore di una etichetta e , se presente, visitare il sottoalbero che da essa si diparte.

Considerando l'albero di Figura 15.3, se l'utente immette f , il programma deve visualizzare f, g, h, i, l, m . L'algoritmo di ricerca del Listato 15.5 si comporta come la visita in ordine anticipato, con l'eccezione che se, il valore viene reperito, il procedimento si blocca e ritorna il puntatore al nodo in esame.

/ Da aggiungere al Listato 15.4 */*

```
...
struct nodo *trovato;
char sotto_albero, invio;
...

scanf("%c", &invio);
printf("\nInserisci la radice del sottoalbero: ");
scanf("%c", &sotto_albero);

trovato = NULL;
ricerca( radice, sotto_albero, &trovato );
```

```

if(trovato!=NULL) {
    printf("\n SOTTOALBERO IN ORDINE ANTICIPATO \n");
    anticipato( trovato );
}
else
    printf("\n Sottoalbero non presente");
}

/* Visita in ordine anticipato, ricercando il sottoalbero con
radice sa. Se lo reperisce assegna il suo indirizzo a *punt_sa */

ricerca(struct nodo *p, char sa, struct nodo **punt_sa)
{
if(sa == p->inf)
    *punt_sa = p;
else
{
    p = p->p_arco;
    while(p!=NULL) {
        ricerca(p->figlio, sa, punt_sa);
        p = p->p_arco;
    }
}
}
}

```