

Possiamo modificare la ricerca binaria nel seguente modo, ottenendo così la cosiddetta *ricerca uniforme*:

```
public static int ricercaUniforme (double d, VettoreDouble V, int in,
    int fin, Ord ord)
{
    // test più sofisticato per la risoluzione diretta
    if ( in >= fin || ord.precede(a,V.valDouble(in)) ||
        ord.precede(V.valDouble(fin),d) ||
        ord.uguale(V.valDouble(in),V.valDouble(fin)) )
        return (in == fin && ord.uguale(d,V.valDouble(in)))?
            in: V.iMax()+1;
    else
    {
        // individuazione più accurata dell'unico sottoproblema
        int x = Math.round(in+(fin-in)*(d - V.valDouble(in))/
            (V.valDouble(fin)-V.valDouble(in)));
        if ( ord.precede(V.valDouble(in),d) )
            in = x+1;
        else
            if ( ord.precede(d,V.valDouble(in)) )
                fin = x - 1;
            else
                in = fin = x;
        // risoluzione dell'unico sottoproblema e
        // utilizzo della sua soluzione
        return ricercaUniforme(d,V,in,fin,ord);
    }
}
```

Il test più sofisticato per verificare se procedere alla risoluzione diretta del problema poteva essere utilizzato anche nella ricerca binaria; nel caso della ricerca uniforme, esso diventa obbligatorio per evitare di avere valori negativi o addirittura divisioni per zero nel calcolo di x . Al contrario della ricerca binaria, la ricerca uniforme è un algoritmo naturale che spesso utilizziamo nel consultare un dizionario o un elenco telefonico. Infatti, se dobbiamo ricercare una parola iniziante con "S", andremo verso la fine dell'elenco mentre, se la parola inizia con "B", andremo verso l'inizio e andremo al centro solo nel caso la parola inizi con una delle lettere centrali dell'alfabeto.