

```
1 (* Applichiamo Programmazione Strutturata allo Sviluppo del QuickSort.
2   Vincoli:
3   + Una sequenza c;
4   + di valori ordinabili t (generico)
5   + Quindi, c e' un valore di tipo Seq(t), sequenze di tipo t.
6   + Ordinamento totale op:
7     perogni x,y: (1) (op x y)or(op y x); (2) (op x y)and(op y x) iff x=y;
8   Algoritmo
9   1) Sia cc la sequenza corrente da ordinare;
10  2) Sia op ordinamento totale sui valori di tipo t da utilizzare;
11  3) Sia pv di tipo t, un elemento di separazione per cc;
12  4) Dividiamo cc in due sottosequenze cLE, cGT separate da a;
13  5) Ripetere 1)-5) su ogni sottosequenza, non singoletta, ottenuta da 4);
14  6) Terminiamo presentando la sequenza ottenuta
15  Osservazioni:
16  o1: pv elemento di cc, arbitrariamente scelto e rimosso da cc;
17  o2: separazione:
18      + se x in cLE allora (op x a)or(x=a)
19      + se x in cGT allora (op a x)
20      + cLT seguito da pv, seguito da cGT, e' una permutazione di cc
21  o3: op e' un valore funzione (predicato) trasmesso come parametro.
22 *)
23
24 # use "SeqT.ml";; (* carica il codice del tipo di dato:'t seq *)
25                  (* per sequenze polimorfe in 't          *)
26
27 let rec quickSort (cc:'t seq) (op:'t ->'t ->bool) =
28   (* Out: copia ordinata con op degli elementi di cc *)
29   (* effect: nessuna modifica                               *)
30   if size cc <= 1 then cc
31   else let (pv,cv) = pivot cc in
32         let (cLE,cGT) = dividi cv op pv in
33         concatena (quickSort cLE op)(cons pv (quickSort cGT op))
34 ;;
35
36
37
38
39 (* quando importiamo SeqT,ml:
40 # let cc = Cons(2,Cons(5,Cons(0,Cons(-3,Cons(0,Cons(21,Cons(0,Cons(2,Cons(0-2,Cons(0,Cor
41 val cc : int seq =
42   Cons (2,
43     Cons (5,
44       Cons (0,
45         Cons (-3,
46           Cons (0,
47             Cons (21, Cons (0, Cons (2, Cons (-2, Cons (0, Cons (3, Nil))))))))))val cc : ir
48 # val cc : int seq =
49   Cons (2,
50     Cons (5,
51       Cons (0,
52         Cons (-3,
53           Cons (0, Cons (0, Cons (2, Cons (-2, Cons (0, Cons (3, Nil))))))))))
54     Cons (21, Cons (0, Cons (2, Cons (-2, Cons (0, Cons (3, Nil))))))))))cc;;
55 Error: Syntax error
56 # cc;;
57 - : int seq =
58 Cons (2,
59   Cons (5,
60     Cons (0,
```

```
61     Cons (-3,  
62     Cons (0,  
63     Cons (21, Cons (0, Cons (2, Cons (-2, Cons (0, Cons (3, Nil))))))))))  
64 # quickSort cc (<);;  
65 - : int seq =  
66 Cons (-3,  
67     Cons (-2,  
68     Cons (0,  
69     Cons (0,  
70     Cons (0,  
71     Cons (0, Cons (2, Cons (2, Cons (3, Cons (5, Cons (21, Nil))))))))))  
72 #  
73 *)  
74  
75 (* quando importiamo SeqT1.ml  
76 # #use "QuickSortT.ml";;  
77 type 't seq = 't list  
78 val size : 't seq -> int = <fun>  
79 val pivot : 't seq -> 't = <fun>  
80 val triPartition :  
81     't seq -> ('t -> 't -> bool) -> 't -> 't seq * 't seq * 't seq = <fun>  
82 val concatena : 't seq -> 't seq -> 't seq = <fun>  
83 val cons : 't -> 't seq -> 't seq = <fun>  
84 val quickSort : 'a seq -> ('a -> 'a -> bool) -> 'a seq = <fun>  
85 # quickSort [2;5;0;-3;0;21;0;2;0-2;0;-3] (<);;  
86 - : int seq = [-3; -3; -2; 0; 0; 0; 0; 2; 2; 5; 21]  
87 *)
```