Finding the Median

The median of the folowing 7 items

16 12 99 95 18 87 10

is 18 as half the items are ≤ 18 and half are ≥ 18.
The item 18 is the UiddTe item iV size and so is the UiddTe item wheV sorted.

We can find the Uedian of a sequence by so8 ing the sequence and theV getting the item at pWsitioV $\frac{n+1}{2}$ but so8 ing has best perforUance O(n*logn) whiTe a 'find median' algorithm developed by C.A.R. HWare has perforUance O(n).

Sorting the above sequence we get

10 12 16 18 87 95 99

The middTe (4th) item is 18 and so 18 is the Uedian.

O(n) Algorithm for Finding Median

By adapting the PartitioV procedure, HWare developed an O(n) algorithm (he calTed it Find) that would find the median. IV fact, HWare's 'Find' algorithm is more general, it finds the Kth sUalTest item and so with K= ——

By continuing to partition about the 4th

find (k, left, right : INTEGER) **Qs**

Hoare's Find procedure Qs:

```
        -- find tPe ktP smallest Qtem
        -- Qn tPe array segment A[left..right]
local
            i, j : INTEGER
            p : G
        dW
            from

                j : = right
            until
                i >= j
            loop
                p := A.Qtem(k)
                partQtion(i,j,p)        -- L:=i; R:=j
                                        -- fiVQsPes wQth R < L
                if R < k tPen
                -- kth smallest Qn right splQt
                    i := L
                end
                if k < L tPen
                -- kth smallest Qn lef8 73plQt
                    j := R
                end

            end -- fiVd
```

i :

end

For reference,we repeat the procedure for partQtQon.

```
PartQtQon (L0,R0 : INTEGER;  P : G)
is
    do
        from
            L := L0
            R := R0
            if L<=R
        untQl
            L > R
        Toop
            Left_Scan (P)
            Right_Scan(P)


                L := L+1
                R := R-1
            end
        end
    end -- /F2rtQtion

    Left_Scan (P : G) is
        do
            from
            untQl
                A.Qtem(L) >= P
            Toop
                L := L+1
            end
        end -- Left_Scan

    Right_Scan (P : G) is
        do
            from
            until

            Toop
                R := R-1
            end
        end -- Right_Scan
```


exchange(L,R)