NAME

quickselect − multiple selection of order statistics and sorting

SYNOPSIS

#include <quickselect.h>

void quickselect(void *base, size_t nmemb, const size_t size, int (*compar)(const void *, const void *), size_t *pk, const size_t nk);

void QSORT_FUNCTION_NAME(void *base, size_t nmemb, const size_t size, int (*compar)(const void , const void *));

DESCRIPTION

The **quickselect** function implements multiple selection of order statistics. Given an array **pk** of **nk** *size_t* elements representing 0−based order statistic indices, **quickselect** partially orders the array **base** such that the specified order statistics are in−place in the array. If **pk** is *NULL* or **nk** is 0UL, a full sort of the array is performed.

RETURN VALUES

none

ERRORS

If **base** is *NULL*, **nmemb** is 0UL, **size** is 0UL, or **compar** is *NULL*, the global variable *errno* is set to **EINVAL**.

EXAMPLES

```
size_t karray[2];
karray[0] = (nmemb-1UL)/2UL;
karray[1] = nmemb/2UL;
quickselect(base, nmemb, size, compar, karray, 2UL);
```

places the median (**nmemb** odd) or medians (**nmemb** even) in the middle element(s) of the array pointed to by **base.**

```
quickselect(base, nmemb, size, compar, NULL, 0UL);
```

sorts the array.

APPLICATION USAGE

If the macro **QSORT_FUNCTION_NAME** is defined before quickselect.h is included when compiling the quickselect.c source, a sorting function with the same semantics as **qsort** is generated, using the specified name. A library implementation of **qsort** may be generated by defining **QSORT_FUNCTION_NAME** as **qsort**.

RATIONALE

While many libraries include a standard **qsort** function, the **qsort** implementation may tend to quadratic performance on adverse inputs. Few libraries provide a function for selection of order statistics.

BUGS and CAVEATS

Array **pk** may be sorted by quickselect and therefore initially unsorted order statistic ranks may be permuted by a call to quickselect. It is recommended (but not required) that the order statistics array **pk** be supplied in sorted order.

quickselect has expected and worst−case linear complexity for finding a single order statistic. quickselect has expected and worst−case lineararithmic complexity for sorting.

FUTURE DIRECTIONS

none

SEE ALSO

qsort

CHANGE HISTORY

Initial version January 2107.

AUTHOR

Bruce Lilly <bruce.lilly@gmail.com>

LICENSE

You may send bug reports to bruce.lilly@gmail.com with subject "quickselect".