|  |
| --- |
| 0hnQGAuNPo4wojnr77c2x8cWiaT70FGHBZ2h-3__FYQ.png |
| Quicksort |
| Data Structures Made Easy |
|  |

Dublin City university

# 1. *Quicksort*

class quick\_Sort{

private static void swap(int [] quick\_Array, int low, int high){

int temporary = quick\_Array[low];

quick\_Array[low] = quick\_Array[high];

quick\_Array[high] = temporary;

}

public static void quick(int [] quick\_Array, int low, int high){

int lo = low;

int hi = high;

if (lo >= hi)

return;

int pivot = quick\_Array[(lo + hi) / 2];

while (lo < hi) {

while (quick\_Array[lo] < pivot)

lo++;

while (quick\_Array[hi] > pivot)

hi--;

swap(quick\_Array, lo, hi);

}

if (hi < lo) {

swap(quick\_Array, hi, lo);

}

int element = lo;

if(lo == low)

element++;

quick(quick\_Array, low, lo);

quick(quick\_Array, element, high);

}

public static void main(String [] args){

System.out.print("Enter the number of elements: ");

int size = Console.readInt();

int [] quick\_Array = new int[size];

System.out.print('\n' + "Enter the elements: ");

for(int index\_1 = 0; index\_1 < size; index\_1++){

int element = Console.readInt();

quick\_Array[index\_1] = element;

}

quick(quick\_Array, 0, size-1);

System.out.print('\n' + "The sorted list is: ");

for(int index\_2 = 0; index\_2 < size; index\_2++)

System.out.print(quick\_Array[index\_2] + " ");

}

}