Exercises — Selection sort

version #

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

Contents

*https://intra.assistants.epita.fr

1 Selection sort

Files to submit:

- selection_sort/selection_sort.c

Authorized headers: You are only allowed to use the functions defined in the following headers:

- stddef.h
- assert.h
- errno.h
- err.h

1.1 Goal

The selection sort is a basic, straigthforward, sorting algorithm. It iterates over the array, say A, and, for each index i, finds the element at index j such that $j > i$ and $A[j] < A[i]$. If such index j exists, it swaps $A[i]$ and $A[j]$. Otherwise stated, for each iteration, it looks in the remaining array if it can find an element smaller than the current one and swaps them.

$$\forall i, \text{ swap } A[i] \text{ with } \begin{cases} A[j] & \text{if } \exists j > i \text{ such as } A[j] < A[i] \\ A[i] & \text{otherwise} \end{cases}$$

Selection sort is a good introduction to sorting algorithms because of its simplicity, but always keep in mind that its time complexity is *quadratic* ($O(n^2)$).

First, you have to code a function that returns the index of the smallest element in an array. The starting index and the size of the entire array are given.

```
unsigned array_min(const int arr[], unsigned start, unsigned size);
```

Using the previous function, you have to code the selection sort. `arr` is the array to be sorted and its size is given.

```
void selection_sort(int arr[], unsigned size);
```

All given arguments are considered valid.

1.2 Example

```
#include <stdio.h>

unsigned array_min(const int arr[], unsigned start, unsigned size);
void selection_sort(int arr[], unsigned size);

int main(void)
{
    const unsigned size = 35;
```

(continues on next page)

(continued from previous page)

```
    int arr[] = {
        5,   8,   90,   3,    7,    64,   10224, 88,   39,    78,   20,   6,
        9,   79,  30,   45,   908,  201,  73,    460,  1330,  37,   32,   13,
        709, 310, 1998, 2000, 2020, 2021, 2022,  5600, 10000, 4560, 4800,
    };

    unsigned expected[] = {
        3,  3,  3,  3,  11, 11, 11, 11, 11, 11, 11, 11, 12, 23, 23, 23, 23, 23,
        23, 23, 23, 23, 23, 23, 25, 25, 26, 27, 28, 29, 30, 33, 33, 33, 34,
    };

    for (unsigned i = 0; i < size; ++i)
        printf("%u: %d = %d\n", i, array_min(arr, i, size), expected[i]);

    printf("Before sorting:");
    for (unsigned i = 0; i < size; i++)
        printf(" %d", arr[i]);

    selection_sort(arr, size);

    printf("\nAfter sorting:");
    for (unsigned i = 0; i < size; i++)
        printf(" %d", arr[i]);

    return 0;
}
```

```
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic -o sort selection_sort.c main.c
42sh$ ./sort
0: 3 = 3
1: 3 = 3
2: 3 = 3
3: 3 = 3
4: 11 = 11
5: 11 = 11
6: 11 = 11
7: 11 = 11
8: 11 = 11
9: 11 = 11
10: 11 = 11
11: 11 = 11
12: 12 = 12
13: 23 = 23
14: 23 = 23
15: 23 = 23
16: 23 = 23
17: 23 = 23
18: 23 = 23
19: 23 = 23
20: 23 = 23
21: 23 = 23
22: 23 = 23
23: 23 = 23
```

(continues on next page)

(continued from previous page)

```
24: 25 = 25
25: 25 = 25
26: 26 = 26
27: 27 = 27
28: 28 = 28
29: 29 = 29
30: 30 = 30
31: 33 = 33
32: 33 = 33
33: 33 = 33
34: 34 = 34
Before sorting: 5 8 90 3 7 64 10224 88 39 78 20 6 9 79 30 45 908 201 73 460 1330 37 32 13 709 310 1998 2000 2020 2021 2022 5600 10000 4560 4800
After sorting: 3 5 6 7 8 9 13 20 30 32 37 39 45 64 73 78 79 88 90 201 310 460 709 908 1330 1998 2000 2020 2021 2022 4560 4800 5600 10000 10224
```

It is my job to make sure you do yours.