ESE-2025 Lab 1

Name: Shreya Mamadapur Student ID: 774035
Instructor: Takis Zourntos

Introduction

In this program, we sort an array in ascending order using bubble sort. The sorting function, `bsort() takes` a pointer to the array and the size of the array as parameters. And using pointers it writes back the sorted values to the actual array passed on to the function.
Bubble sort works by comparing the adjacent values and swapping them left or right(based on ascending or descending sort). This swapping process has to be done on all the elements of the array N times, which means that we have to use two loops for sorting the values. This process, although uses basic comparison and swapping operations, can get very inefficient as the number of elements in the array goes high, due to the total number of computations required.
In our program, we use the dispArray() function(similar parameters as bsort()) for displaying the values of the array before and after sorting.
The value of the array is hardcoded in the program and can be seen to change after running the bsort() function.

Conclusion

With the program, we got a better idea of passing pointers to functions and modifying the actual array. Although the bubble sort code is inefficient, it helped understand and appreciate the simple concept of it.

Appendix

Code :

```
/*
============================================================================
Name        : bsort.c
Author      :
Version     :
Copyright   : Copyright (C) Y'all
Description : demonstrates the Bubble Sort Algorithm
============================================================================
*/
```

ESE-2025 Lab 1

```c
#include <stdio.h>
#include <stdlib.h>
/*
 * function to print out the array contents
 */
void dispArray(const int *s, size_t N)
{
      for (size_t i=0; i != N; ++i)
      {
            printf("%d ",s[i]);
      }
      printf("\n");
      return;
}
/*
 * function to sort the array
 */
void bsort(int *s, size_t N)
{
    int temp;
    for(int i=0; i<N-1; i++)
    {
          for(int j=0; j<N-1; j++)
          {
              if(s[j]>s[j+1]) //compare an array element to the next element
              {
                  temp=s[j]; //swap to push the larger number into higher index
                  s[j]=s[j+1];
                  s[j+1]=temp;
              }
          }
      }
}

int main(void)
{
      int s[]={22,32,9,0,1,2,45,668,932,26,5,3,333,4,7}; // our data array
      size_t N=sizeof(s)/sizeof(int); // number of elements in s
      printf("the original data is:\n");
      dispArray(s, N);
```

```
	bsort(s, N);
	printf("the sorted data is:\n");
	dispArray(s, N);
	printf("bsort program complete.\n\n");

	return EXIT_SUCCESS;
}
```