Algorithms and Data Structures – Sheet 3

Valdrin Smakaj

Problem 1 is attached in the other pdf called pr1

Problem 2.

2.1) Implement Selection sort

~~~Answer~~~

```
void selecSort(int A[],int n){
    for(int i=0; i<n-1; i++){
        int nav=i,x;
        for(int j=i+1; j<n; j++){
            if(A[j]<A[i]){
                nav=j;
            }
        }
        if(nav!=i){
            x=A[nav];
            A[nav]=A[i];
            A[i]=x;
        }
    }
}
```

2.2) Show that Selection Sort is correct (Hint: consider the loop invariant).

~~~Answer~~~

We consider the inner and outer loop variants from the beggining, during the loop iteration and in the termination place.

A) Outer loop
   a) Begginning- Take the first element but since its in the beginnign, it is empty and therefore the array of the data type is empty and already sorted in that place so we continue onto the next element
   b) Through the elements- We go through each element and compare in a loop with the other next elements and if the next element from that position i is less than the element of pos i then then we swap the elements between(which is part of the job that the inner loop does) and leave the pos(i) less than (i+1). And this goes on and on.
   c) Termination position – Since we already sorted each element i less than the position i+1 so in a more understandable way (i(n) < i(n+1) < i(n+2) < i(n+k) for some k>=0) then the

last position of the array is going to remain in the same position since it's the array got sorted already and there is no other smaller value than the one in postion n(last pos) to be swapped with.

B) Inner loop
   a) Beginning – Since we are in the begininng, we are saving the first element in a variable which in this case is currently considered as the small element.
   b) Through the loop iteration – the position of i is being saved as the smallest element but when it enters the for loop then it is compared with each other remaining element after it and therefore if any other element of loop from j(i+1) to n is less than the value of first element of i that we put into our variable, then that variable gets the value of the smaller one and processes on to check if there is any other existing values even smaller than the one it currently got. If it isnt then it means that the first current positon was already the smallest one and it remains in the current postion and now the for loop goes on into the next (i+1) element.
   c) Termination position – Since throughout the loop the elements have been sorted out and the first for loop position iterator has gone to the very last element then if the last element of the array has a smaller element than the one before it then they are swapped using the intermediate variable as a form of swapping as expressed into the code, but if it doesnt, since there are no other elements (n+1) or anything after then the last element will just remain in that position and not be changed with anything else. Therefore in the end the array will be sorted using this algorithm

C) Generate random input sequences of length n as well as sequences of length n that represent Case A and Case B for the Selection Sort algorithm. Case A: the case which involves the most swaps (Hint: it is not a decreasingly ordered array). Case B: the case with the least swaps. Briefly describe how you generated the sequences (e.g., with a random sequence generator using your chosen language)

~~~Answer in the code for C and D~~~

E) If we see there is not really a big difference sice whatever we are doing is just compaing elements from beggining to the end (n-1) and after still n-1-1 so on and on we, putting this in a formula form we'll get:

$$(n-1)+(n-2)+\ldots+1=\frac{n(n-1)}{2}=O(n^2)$$
~~~