

CS101 – Computer Programming
Quiz for Wednesday – 24 September 2014

Q1. Consider the following program written by a student for sorting array in descending order using a slight change in the selection sort algorithm as shown below. Assume that the array A needs to be sorted and the function 'selectionsort' is called from the main program in the form 'selectionsort(A,n)', where 'A' is an array of size 'n'.

```
int findIndexOfMax(int A[],int start, int end){
    int i, currMaxIndex = start;
    for(i = start; i< end; i++){
        if(A[i] >= A[currMaxIndex]) { currMaxIndex = i; }
    }
    return currMaxIndex;
}

int findIndexOfMin(int A[],int start, int end){
    int i, currMinIndex = start;
    for(i = start; i>= end; i--){
        if(A[i] < A[currMinIndex]) { currMinIndex = i; }
    }
    return currMinIndex;
}

void swap(int A[], int index1, int index2){
    if(index1!=index2){
        int temp = A[index1];
        A[index1] = A[index2];
        A[index2] = temp;
    }
}

void selectionsort(int A[100],int n){
    int count;
    int currTop, currMaxIndex, currMinIndex;
    for(currTop = 0; currTop < n/2; currTop++) {
        currMaxIndex = findIndexOfMax(A,currTop,n/2);
        swap(A,currTop,currMaxIndex);
    }
    for(currTop = n-1; currTop <= n/2; currTop--) {
        currMinIndex = findIndexOfMin(A, currTop, n/2);
        swap(A,currTop, currMinIndex);
    }
}
```

Select an appropriate choice with respect to the above code for any arbitrary unsorted Array 'A' as input to the function 'selectionsort':

- A. The above code for selection sort will sort the whole array 'A' in descending order
- B. The above code for selection sort will sort part of the array 'A' in ascending order
- C. The above code for selection sort will sort part of the array 'A' in descending order
- D. None of these

Q2. Consider the following main program for sorting the array A in descending order using selection sort. The call to the function 'findIndexOfMax(A, currTop, n)' returns the index of the maximum element of the array A from 'currTop' to 'n-1'. The call to the function 'swap(A, currTop, currMaxIndex)' swaps an element of the array 'A' located at 'currTop' with that located at 'currMaxIndex'.

```
int main() {  
    // Code for declarations of variables used and reading  
    elements of array A of size n  
    int currTop, currMaxIndex;  
    for (currTop = 0; currTop < n; currTop++) { //main loop  
        currMaxIndex = findIndexOfMax(A, currTop, n);  
        swap(A, currTop, currMaxIndex);  
    }  
    // Rest of code  
    return 0;  
}
```

Here, actual swaps occur when swapping indices are not the same, i.e., $\text{currTop} \neq \text{currMaxIndex}$. Let us define the following quantity called useful work (Useful Work) for an array as the ratio of the number of actual swaps performed to the total number of swaps.

- A. If $A = \{99, 97, \dots, 1, 100, 98, \dots, 4, 2\}$, then $UW = 0.5$
- B. If $A = \{1, 2, \dots, 100\}$, then $UW = 0$
- C. If $A = \{100, 99, 98, \dots, 1\}$, then $UW = 0.5$
- D. None of these