



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

## COS 212 Tutorial 9: Version A

- 03/05/2012
  - 50minutes
  - 3 questions for a total of 20 marks.
- 

Name: \_\_\_\_\_

Student/staff Nr: \_\_\_\_\_

Marker (office use): \_\_\_\_\_

**Question 1** Insertion Sort: ..... (7 marks)

1.1 Suppose an array containing  $n$  elements initially. Comment on the following statement: (2)

If the number of elements is tripled, then the effort of the algorithm will increase by a factor of 9.

**Solution:** The algorithms' complexity is calculated in terms of the number of comparisons which is in turn determined by the number of elements. Insertion sort is an  $O(n^2)$  algorithm so the effort doubles for each doubling of the number of elements, so if  $3n$  then  $\frac{(3n)^2}{n^2} = 9$

Answer:

1.2 Another version of the insertion sort algorithm is given by the following pseudo code:

```
insertion(arr [])
  i = arr.length - 1
  j = 0
  while(i > 1)
  {
    el = arr[i]
```

```

    for (j = 1; j < i && el < arr[j-1]; --j)
        arr[j-1] = arr[j]
    arr[j] = el
    --i
}

```

- a) Highlight the problem(s) with this version of the algorithm.

(3)

Answer:

**Solution:** This version of the algorithm tries to shift elements from left to right. It will overwrite elements. Furthermore, j is decremented instead of incremented which means that the nested could become an infite loop and run out of bounds.

BUT!!!!

I made a mistake in the other tut, to even out the tuts you can give full marks here.

- b) Comment on the stopping condition of the outer **while**-loop.

(2)

**Solution:** Use 1 instead of 0 because the last two elements considered should not be at indices 0 and -1.

Answer:

## Question 2 Selection Sort: ..... (10 marks)

- 2.1 Insertion sort suffers from a serious problem, selection sort attempts to address this problem. Discuss the problem as well as how selection sort tries to remedy it.

(2)

**Solution:** Insertion shifts elements up in the array which requires copying for each iteration. It requires that items be moved. Selection works with indices and only swaps two elements (if needed) in each outer for loop iteration (not many elements shifted)

Answer:

- 2.2 Consider the array `[a,b,c]` where `a,b` and `c` are variables and not values. If no assumptions are made about the values taken on by these variables (i.e. they could be anything) then it becomes invalid to assume, for example, that `a` will have a value less than the value of `b` or `c`. With this in mind, draw the decision tree for this array (8)

**Solution:** See separate figure in this folder.

Give 1 mark for each correct leaf node, they also have to be in order...

Answer:

**Question 3** Bubble Sort: ..... (3 marks)

- 3.1 Consider the following suggested alternative for the bubble sort algorithm: (3)

```
bubblesort(arr [])
{
    for(i = 0; i < arr.length; ++i)
        for(j = arr.length; j > 0; --j)
```

```
    if (arr[j] < arr[j-1])  
        swap(arr[j], arr[j-1])  
}
```

You may assume that `swap` correctly exchanges `arr[j]` and `arr[j-1]`. Discuss this version of the algorithm. In your discussion you must comment the result that this algorithm will generate as well as briefly discussing its complexity.

**Solution:** It will bubble the smaller elements forward, placing them in their correct positions first and the larger elements will by implication find their positions. the inner loop goes through the entire array for every iteration of the outer loop making this version always perform  $O(n^2)$  (this is the best, average and worst case)