Advanced Data Structures, the 1st Assignment

Submit your code and report (report doesn't have to contain anything about code) through Ninova. Any question: ataka@itu.edu.tr

Deadline: July 5, 2011 23:00 (please don't send your hw via e-mail)

Q1) Find the running times of following code pieces. Please specify your answers briefly (7 points each). Answers without description will be ignored.

```
a. for i=1 to N1
        m ← i*i
        for j=1 to N2
            m←m+j
        endfor
        for k=i to N3
            m←m-1
        endfor
   endfor

b. sum←0
   if N1>N2
        for i=N2 to N1
            sum ← sum + i^3
        endfor
   else
        for i=N2 down to N1
            sum ← sum + i^3
            sum ←sum – i
        endfor
   endif

c. N1 ← size of integer array 'arr'// will not be used, just fyi
   vals: empty integer array
   inds: empty integer array
   foreach element of arr
        N2 ← size of integer array 'vals'
        i ← 1
        for i=1 to N2
            if vals[i]<arr[i]
                break
        endfor
        insert arr[i] at vals[i]
        insert i at inds[i]
   endforeach
```

Q2) Represent the running times which you computed at Q1 with big-Oh notation (5 points). Include the value of 'c' which comes from '$|f(x)| \leq c*|g(x)|$' into your answer.

Q3) For the following code piece find (i) the best, (ii) the worst and (iii) average running times. Then represent these running times with one of the asymptotic notations (14 points).

```
var flag: boolean
…
…     // some code here
…     // flag is modified several times
…
var sum: integer
sum ← 0
if flag is true
      for i:1 to N
            sum ← sum + i^2
      endfor
      flag ← true
else
      for i:1 to N
            for j:i to N
                  sum ← sum + i^3
            endfor
      endfor
      flag ← false
endif
```

Q4) Implement the recursive algorithm for solution of n-queens problem using C or C++. According to this problem, 'n' queens can be placed into 'nxn' chess board except for n=2 and n=3.Recursive algorithm for solution of this problem is based on backtracking. According to backtracking idea, queens are placed on the chessboard if there are compatible places. Otherwise last queen that was placed on the chess board is taken back (60 points).

- Find the running time and big-Oh representation of your implementation
- Run your implementation for n=8, n=16, n=32, n=64 and n=128. Draw the graphic of running time versus n.
- Compare the characteristic of the graphic with your running time calculation.

Your program should take 'n' as parameter from command line then it should print the number of solutions to the screen. Be sure that your source code can be compiled with gcc or g++ without any problem.

Hint & Warning: You can find several data structure books focused on this problem in Mustafa Inan Library. Also there are a lot of resources on the internet. That's why this question is not hard to deal with. However exact matches and similarities among submitted codes or between submitted codes and internet resources will be treated as plagiarism. In addition, it is impossible for two students to submit codes with 90% similarity, if they do their homework on their own.