Gebze Technical University
Department of Computer Engineering
CSE 321 Introduction to Algorithm Design
Fall 2016
Midterm Exam
November 16th 2016

Student ID and Name	Q1 (20)	Q2 (20)	Q3 (20)	Q4 (20)	Q5 (20)	Total

Read the instructions below carefully

- All cases of confirmed cheating will be reported for disciplinary action.
- You have 120 minutes.

Q1.

a) Consider a variation of sequential search that scans a list to return the number of occurrences of a given search key in the list. Analyze its best case, worst case, and average case complexities. Is it different from linear search? Why? **(10 points)**

b) Design another algorithm for the same problem, but this time using binary search. Write its pseudo-code. **(10 points)**

(a) Your book by Anany & Levitin, Exercise 2.1, Q3.
Answer is on the Internet.

(b) Algorithm BinaryNumOccurrences (input, x, n)
searched key
input size

i ← firstOccur(input, 0, x, n) // Use binary search to get the first occurrence of x

j ← lastOccur(input, i, x, n) // Use binary search to get the last occurrence of x

return j − i + 1

CONT'D

```
procedure firstOccur( input, index, x, n)
  if (n ≥ index)
      mid ← (n+index)/2
      if (mid=0 OR ( x>input[mid-1] AND input[mid]=x)
          return mid
      else if ( x>input[mid])
          return firstOccur( input, mid+1, x,n)
      else
          return firstOccur(input, mid-1, x,n)

procedure lastOccur (input, index, x,n)
  if (n ≥ index)
    mid ←(n+index)/2
    if (mid =n OR ( x< input[mid+1] AND input[mid]=x)
        return mid
    else if (x < ~~a~~ input[mid]
        return lastOccur(input, ~~index mid-~~ mid-1, x,n)
    else
        return lastOccur(input, mid+1, x,n)
```

Q2. List the following functions according to their order of growth from the lowest to the highest. Prove the accuracy of your ordering. **(20 points)**

Note: Merely stating the ordering without providing any mathematical analysis will not be graded!

a) 3^n

b) $\sqrt[3]{n}$

c) $\ln^2(n)$

d) $(n-2)!$

e) 2^{2n}

Your book, by Anany & Levitin,
Exercise 2.2, Q5
Answer is on the Internet.

Q3. Design a BFS-based algorithm to check whether a given graph is bipartite. Analyze the complexity of your algorithm using big-Oh notation. **(20 points)**

(*) Explained in class...

Algorithm BipartiteBFS (G)

s ← Pick a random vertex

foreach u ∈ V[G] \ {s}

do color[u] ← ~~white~~ red

d[u] ← ∞

partition[u] ← 0

end for

color[s] ← ~~black~~ blue

partition[s] ← 1

d[s] ← 0

Queue Q ← [s]

while Q is not empty

do u ← head[Q]

foreach v ∈ Adj[u] do // For each v in the neighborhood of u

if partition[u] ← partition[v]

then return 0

else

if color[v] ← red then

color[v] ← blue

d[v] ← d[u]+1

partition[v] ← 3-partition(u)

enqueue(Q, v)

Dequeue(Q)

Color[u] ← white

Return 1

Q3. Design a BFS-based algorithm to check whether a given graph is bipartite. Analyze the complexity of your algorithm using big-Oh notation. **(20 points)**

(*) Explained in class...

Algorithm BipartiteBFS (G)

s ← Pick a random vertex

foreach u ∈ V[G] \ {s}

do color[u] ← ~~white~~ red

d[u] ← ∞

partition[u] ← 0

end for

color[s] ← ~~black~~ blue

partition[s] ← 1

d[s] ← 0

Queue Q ← [s]

while Q is not empty

do u ← head[Q]

foreach v ∈ Adj[u] do // For each v in the neighborhood of u

if partition[u] ← partition[v]

then return 0

else

if color[v] ← red then

color[v] ← blue

d[v] ← d[u]+1

partition[v] ← 3-partition(u)

enqueue (Q, v)

Dequeue(Q)

Color[u] ← white

Return 1

Q4. Design an exact decrease-and-conquer algorithm for the following task by writing its pseudocode: For any even n, mark n cells on an infinite sheet of graph paper so that each marked cell has an odd number of marked neighbors. Two cells are considered neighbors if they are next to each other either horizontally or vertically but not diagonally. The marked cells must form a contiguous region, i.e., a region in which there is a path between any pair of marked cells that goes through a sequence of marked neighbors. **(20 points)**

Your book, Anany & Levitin
Exercise 4.1, Q3
Solution is on the Internet.

Q4. Design an exact decrease-and-conquer algorithm for the following task by writing its pseudocode: For any even n, mark n cells on an infinite sheet of graph paper so that each marked cell has an odd number of marked neighbors. Two cells are considered neighbors if they are next to each other either horizontally or vertically but not diagonally. The marked cells must form a contiguous region, i.e., a region in which there is a path between any pair of marked cells that goes through a sequence of marked neighbors. **(20 points)**

Your book, Anany & Levitin

Exercise 4.1, Q 3

Solution is on the Internet.

Q5. Write the pseudocode of linear search with repeated elements. Analyze its best case, worst case, and average case complexities.

Your HW question.
Solutions are already announced.