Assignment 01

Instructor: Mehrdad Nojoumian
Course: Data Structure and Algorithm Analysis

Deadline: Oct 11

1. Write Pseudocode for the following
   a. Write pseudocode to determine if there are more odd or even numbers in a list of integers. It should take a list of integer numbers as input and return either “odd” or “even”. For example, if given the list [1,2,3,4,5] as input your pseudocode should return “odd”

   b. Cosine similarity (cos sim) is a commonly used similarity metric that measures how similar two vectors (arrays) of numbers are. The equation for cosine similarity is given by

$$sim(\boldsymbol{A},\boldsymbol{B}) = \frac{\boldsymbol{A}*\boldsymbol{B}}{||\boldsymbol{A}||*||\boldsymbol{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt[2]{\sum_{i=1}^{n} A_i^2}\sqrt[2]{\sum_{i=1}^{n} \boldsymbol{B}_i^2}}$$

   where A_i and B_i are components of vector **A** and **B** respectively. Write pseudocode to calculate cosine similarity of two vectors **A** and **B**. it should take two arrays of numbers, **A** and **B**, as arguments and return a single value, their cosine similarity.

2. Big-O analysis (Worst Case Time Complexity)

   a. Find the worst case runtime (big-O notation) for the following pseudo code which returns true if an integer n is prime, false if it is not prime.

```
procedure isPrime(n)
  if n <= 1
    return false
  else if n <= 3
    return true
  else
    for i in 2 to sqrt(n)
      if n % i == 0
        return false
    return true
```

   b. Find the worst case runtime (big-O notation) for the SelectionSort algorithm. This algorithm sorts an array A of length n.

```
proceedure SelectionSort(array a, length(A) = n)
  for i in 0 to n - 2
    maxIndex = i
    for j in (i + 1) to (n - 1)
      if a[j] > A[maxIndex]
        maxIndex = j
      swap(A[i], A[maxIndex])
```

c. Find the worst case runtime (big-O notation) for BogoSort, pseudocode is given below:

```
procedure BogoSort(array A)
  while not isInOrder(A):
      shuffle(A)
```

shuffle() randomly reorders A. isInOrder() returns true if the list is in order, false otherwise. It has the following pseudocode:

```
procedure isInOrder(array A, length(A) = n)
  for i in 0 to n-1
    if A[i] > A[i+1]
      return false
```

3. Find the average runtime complexity of binary search

```
procedure binary search ($x$: integer, $a_1, a_2, \ldots, a_n$: increasing integers)
  $i := 1$ {$i$ is the left endpoint of interval}
  $j := n$ {$j$ is right endpoint of interval}
  while $i < j$
      $m := \lfloor (i + j)/2 \rfloor$
      if $x > a_m$ then $i := m + 1$
      else $j := m$
  if $x = a_i$ then location := $i$
  else location := 0
  return location
```

4. Greedy strategy

a. Does using the greedy algorithm for making change of n cents give a correct solution if a nickel was worth 6 cents instead of 5? (we have 1, 6, 10 and 25 cent coin denominations).

b. We have n unique items. For i = 1,2,3,…,n, each item has a weight $w_i > 0$ and value $v_i > 0$. Write a greedy algorithm to find the maximum value of items that can be carried in a knapsack with a maximum weight capacity of W. An item may be broken into pieces and only a fraction put into the knapsack. It should take an array of weights, an array of item values and the weight limit of the knapsack as input. The array of weights and items are ordered such that w_i corresponds to the same item as v_i.