Due: 15th October 2024
20% penalty for 1 day late
40% penalty for 2 days late
Submission not allowed afterwards

CS 2009: Design and Analysis of Algorithms
Assignment 2
Total Marks: 100 points

---

**Question 1: 10 points**

HeapSort is a sorting technique that takes O(n logn) time. Watch any video to understand how HeapSort works.

You are given an unsorted array A=[4,1,3,2,16,9,10,14,8,7]

1. Apply the **HeapSort** algorithm to sort the array.
2. Display the array at each major iteration of the heap-building and sorting process (both during the heap construction and after each element is extracted from the heap).

**Question 2: 20 points**

(a) You've learned the binary search algorithm, which is used to search for an element in a sorted array. Another efficient search algorithm is Jump Search, which optimizes search time by skipping multiple elements at once. Learn about the Jump Search algorithm from this link.

- Using any sorted array of size at least 8, perform a search for a number located at the end of the array using both Binary Search and Jump Search.
- Provide a detailed step-by-step breakdown of each iteration of both algorithms, showcasing the decision-making process at each step.
- After completing both searches, compare their efficiency in terms of time complexity and space complexity.
- Discuss in-depth the advantages and disadvantages of both techniques, including when one might outperform the other based on different scenarios such as array size, structure, and the position of the target element.

(b) Interpolation Search and Exponential Search are two additional advanced search algorithms. Study the concepts of both from this link and then:

1. Explain the inner workings of both Interpolation Search and Exponential Search in your own words. Be sure to highlight how they differ from traditional search techniques like Binary Search.
2. Provide a real-world scenario where each algorithm might excel, comparing their performance based on factors like array size, distribution of data, and search element location.
3. For each search algorithm (Interpolation and Exponential), discuss its best-case, average-case, and worst-case time complexity. Include insights into how data distribution (e.g., uniformly distributed or not) can affect their performance.

**Question 3: 20 points**

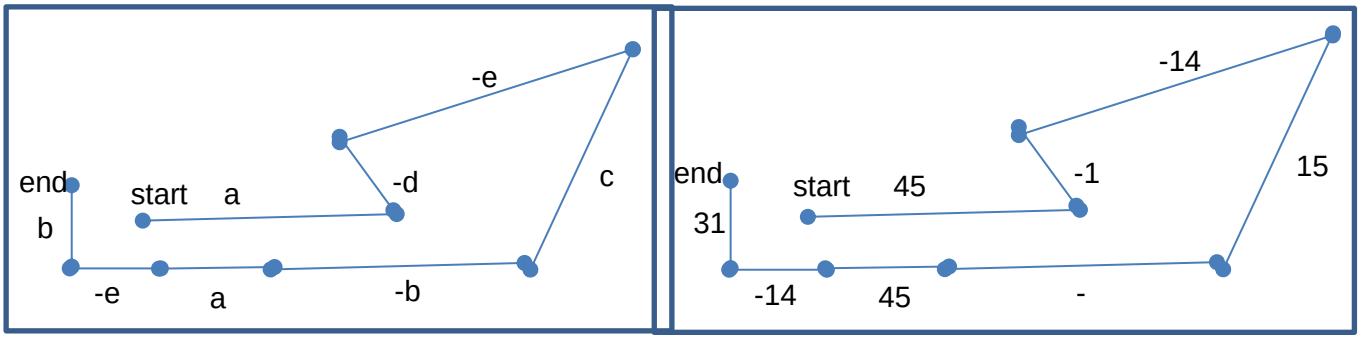
Find the values of the following variables, using your student ID. The values of these variables will be used, for solving the below problem.

a = first two digits of your student ID
b = last two digits of your student ID
c = (multiply your first two digits by 7) mod 60
d = (your complete four digits student ID) mod 30
e = positive ( (last two digits of your student ID) - (first two digits of your student ID) )

For example, if the student ID is 21k-4531, then the value of the variables will be calculated as follows:
a = 45
b = 31
c = (45*7) mod 60 = 315 mod 60 = 15
d = 4531 mod 30 = 1
e = positive (31 – 45) = positive (-14) = 14

**Problem to solve**

A Rickshaw driver moves from one road to another to find customers. If a driver does not find any customer on some road, then only fuel is consumed and thus has a loss equal to price of fuel, represented by negative amount (in figure below). If the driver, finds some customer, then some profit is earned. The rickshaw driver made a route map (as shown below), where he used to travel, and also note down the average benefit earned for the specific roads. The rickshaw driver is now eager to know, what is the highest profit he can get while traversing any contiguous subset of route.




(a) (b)

Figure 1. The Route map

1. Redraw the map in Figure 1(a) using the values, calculated using your student ID

   (Just for understanding, map is redrawn for the example student ID in the Figure 1(b))

2. Using the maximum sub-array algorithm, dry run on your redrawn graph (that is one, having your own calculated values), to get the highest-profit contiguous subset of route.

**Question 4: 20 points**

Find the contents of array C[], using your student ID, then show dry run to sort it in O(n) time. For example, for student ID: 21k-4509, The content of array C = {de, dei, a, e, ei, ba}

The contents of array have to be derived as follows:

- First find the contents of array S[].
- The contents of array S must be of this given pattern: S = {45, 4509, 09, 50, 509, 21}, which is derived for student ID: 21k-4509.
- Keenly observe the pattern, in which the student ID digits are used.
- That is, the first element of array, is the first two digits of student ID, etc.

- For every digit in the above S array, find the corresponding alphabet for the digit value.
- That is, for a digit value, the respective alternate alphabet value will be: 1=a, 2=b, 3=c, 4=d, 5=e, 6=f, 7=g, 8=h, 9=i, (and 0=ignore).
- Finally: the string value for each content of an array S: 45=de, 4509=dei, 01=a, 50=e, 509=ei, 21=ba
- i.e. C={de, dei, a, e, ei, ba}
- (Note! No respective alternate alphabet is used for digit ‘0’)

**Question 5: 10 points**

Given an array, you need to shift all zeros to end of array with out changing relative order of other non-zero elements.

For example If A = [1,0,4,0,0,0,2,0] then output = [1,4,2,0,0,0,0,0]

Design algorithm for this that takes O(n) time and O(1) space.

**Question 6: 10 points**

For the set of keys {1, 4, 5, 10, 16, 17, 21}, draw binary search trees of height 2, 3, 4, 5, and 6.

**Question 7: 10 points**

Find the duplicate element in an array of size 100 in O(n) time.