CSE 321 - Homework 4

Due date: 25/12/2022, 23:59

1. ***20 pts.*** Consider a computer game with a 2D map with axes $A(A_1, A_2, ..., A_n)$ and $B(B_1, B_2, ..., B_m)$. The goal is to start from A_1B_1, move step by step to arrive at A_nB_m, and reach the highest possible score. At each coordinate the player arrives, they gain a (positive) number of game points. Additionally, there is a rule that restricts the movements. If the player is at A_iB_j, their next move should be either A_iB_{j+1} or $A_{i+1}B_j$, and no other movement is possible.

   Example:

   Input: $n = 4, m = 3$
   Game map:

	B_1	B_2	B_3
A_1	25	30	25
A_2	45	15	11
A_3	1	88	15
A_4	9	4	23

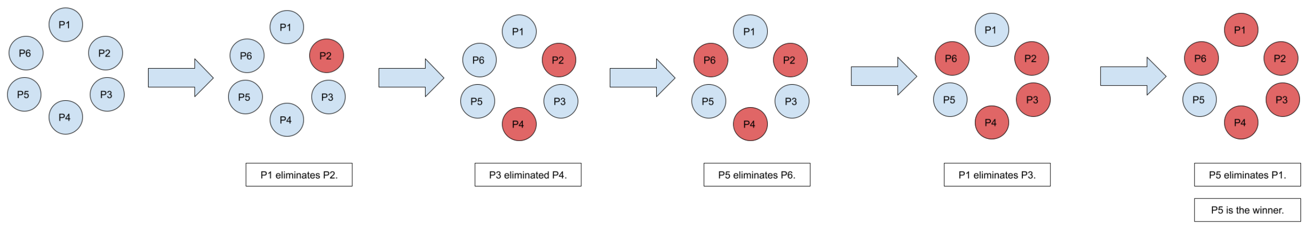
   Output:
   Route: $A_1B_1 \longrightarrow A_2B_1 \longrightarrow A_2B_2 \longrightarrow A_3B_2 \longrightarrow A_3B_3 \longrightarrow A_4B_3$
   Points: $25 + 45 + 15 + 88 + 15 + 23 = 211$

   Design a brute-force algorithm to find the sequence of steps to reach the maximum number of total points.

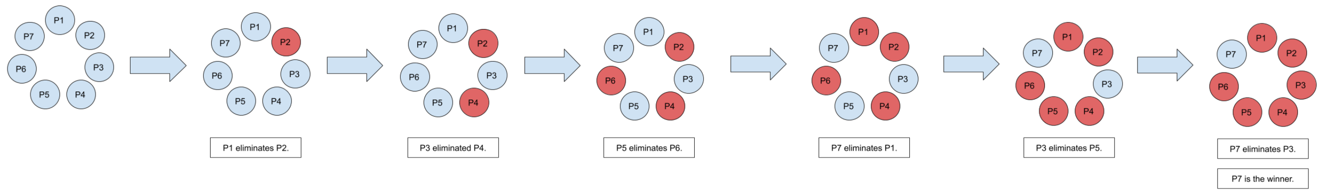
2. ***20 pts.*** Design a decrease and conquer algorithm that finds the median of an unsorted array.

3. Consider a game with n players $\{P_1, P_2, ..., P_n\}$. The players are lined up circularly and at each step, a player eliminates the nearest player on their left. The game starts with P_1's move. P_1 eliminates P_2. Then the next player in the line, P_3, makes a move and eliminates P_4. In the end, only one player is left and that player wins the game. Examine the following examples to understand the game better.

   Example: $n = 6$



Example: $n = 7$



(a) ***10 pts.*** Design an algorithm that finds the winner of the game, by using a circular linked list. Make sure your algorithm runs in linear time.

(b) ***20 pts.*** Design a decrease-and-conquer algorithm that finds the winner of the game. Make sure your algorithm runs in logarithmic time.

4. ***20 pts.*** Ternary search is a search algorithm similar to binary search but it requires the array to be divided into 3 parts instead of 2 parts at each step. The time complexity of ternary search is $O(log_3 n)$ while the time complexity of binary search is $O(log_2 n)$. It seems like there is an improvement in terms of time complexity since $log_3 n < log_2 n$.

   Compare the time complexities of these two algorithms. Explain how the divisor affects the complexity of the search algorithm. Assuming the array has n elements, what does the time complexity of the algorithm become if we divide it into n parts at the beginning?

5. Learn about interpolation search and answer the following questions.

   (a) ***5 pts.*** What is the best-case scenario of interpolation search? What is the time complexity of it?

   (b) ***5 pts.*** What is the difference between interpolation search and binary search in terms of the manner of work and the time complexity?

Important Notes

- For the first 3 problems, implement your solution in Python3. Write a driver function to test each of these algorithms. Inputs should be randomly generated (by using *random* library) or taken from the user (you may assume that the inputs are proper). Gather all of the python code in a single .py file. Do not use external libraries or functions to implement a part of the solution. Pay attention to clean coding.

- Write a report explaining the reasoning behind the algorithms you coded and analyze the worst-case time complexity of each of them. This report should also include your answers to Question 4 and Question 5. Write your report by using a program like MS Office and then convert it to a single PDF file. Pictures of handwritten works are **not accepted**.

- Upload two files only, a .py file and a .pdf file, **not a .zip or a .rar file**.