CS302 - Problem Set 2

Please read the sections of the syllabus on problem sets and honor code before starting this homework.

1. Let `SelfReference` be an algorithm that takes as input a sorted (in increasing order) array A of n distinct (non repeating) positive and negative integers, and returns an index i such that $A[i] = i$, or returns 0 otherwise. (Assume the indices of A start at 1 and go to n.)

   (a) **[9 points]** Write psuedocode for a recursive version of `SelfReference` that is as fast as possible. I've specified the input and output below.

   (b) **[11 points]** Prove your algorithm is correct. (Hint: use strong induction on $n = f - s$.)

   (c) **[6 points]** What is the asymptotic runtime of your algorithm? Give a recurrence relation for the time complexity of your algorithm.

   Algorithm 1: `SelfReference`(A, s, f)

   Input : Sorted (in increasing order) array A of distinct positive and negative integers, a starting index s and an ending index f such that $1 \leq s \leq f \leq length(A)$

   Output: Value i such that $A[i] = i$, and $s \leq i \leq f$, or 0 if no such i exists. (When $s = 1$ and $f = length(A)$ this is standard `SelfReference`)

2. Suppose you have a line graph on 6 vertices with the following weights:

   $$w(v_1) = 3 \qquad w(v_2) = 7 \qquad w(v_3) = 10 \qquad w(v_4) = 5 \qquad w(v_5) = 4 \qquad w(v_6) = 5$$

   (a) **[3 points]** Let G_i be the line graph on the first i vertices with the same weights as above. Let $S(G_i)$ be the max-weight independent set on G_i. What are $S(G_1), S(G_2), \ldots, S(G_6)$? (Note: your answers should be *sets*.)

   (b) **[3 points]** For $i = 3, 4, 5, 6$, verify the recurrence relationship we discussed in class: that if $v_i \in S(G_i)$, then $S(G_i) - v_i = S(G_{i-2})$ while if $v_i \notin S(G_i)$, then $S(G_i) = S(G_{i-1})$.

3. Suppose you are hosting a music festival, and you are trying to decide which set of n possible groups should perform during the prime-time slot. You luckily have an unlimited number of stages, so you can have many different bands play at the same time. The problem is that each band will only perform once, and so if you have several bands play at the same time, a festival attendee who is a huge fan of those bands will be upset that they can't see all of them. To figure out how you should schedule the groups, you give a survey to 100 early-bird registrants, and have them rank the bands that they are most excited to see at the festival.

   (a) **[6 points]** Describe how you could use the algorithm for Max Weight Independent Set to choose which groups should perform in the prime slot. In other words, what should you choose for V, E, and w in your MWIS problem?

   (b) **[6 points]** Given the context of the festival, why is the solution you get from part (a) good?

(c) [**3 points**] Now that you've figured out who should perform in the prime time slot, how should you pick which bands should perform in the second best slot using MWIS?

4. For the closest points problem, we argued we only needed to look at points with x-coordinates that were within $\pm\delta$ of the midline, where δ is the smallest separation found in either recursive call. Explain what range of x-coordinates away from the midline would you need to look at in this step if we instead, throughout the problem, used

   (a) [**6 points**] The Minkowski distance: $(|x_i - x_j|^p + |y_i - y_j|^p)^{1/p}$, for $p \in \mathbb{R}^+$. This distance is what you get on curved spacetime like in general relativity.

   (b) [**6 points**] The skewed distance: $\sqrt{2(x_i - x_j)^2 + (y_i - y_j)^2}$. This distance would make sense in a scenario where it is much harder to travel in the x-direction than in the y-direction. For example, suppose to travel in the y-direction, you can get on highways, but in the x-direction, you have to take local roads...like in Vermont! (Just try to get to Maine from here!)

5. Suppose you have n events, each with a start time s_i and end time f_i, for $i \in \{1, \ldots, n\}$. Unfortunately, you only have one auditorium, and you can't schedule conflicting events (events where a start time of one is between the start time and end time of another.) We would like to maximize the number of events that are held. For each of the following greedy algorithms, create an example of a series of events (with start and end times) where the algorithm does not perform optimally.

   (a) [**3 points**] At each iteration, pick the remaining event with the earliest start time.

   (b) [**3 points**]At each iteration, pick the remaining event that has the shortest time ($f_i - s_i$ is smallest.)

6. Approximately how long did you spend on this assignment (round to the nearest hour)?