Warmup Exercises TravelOptions Programming Assignment

Having trouble getting your head around the concepts of the TravelOptions assignment? Working through the following exercises by hand will hopefully clarify matters. Feel free to compare your answers with classmates on piazza.

Here are two lists of options A and B (format: <price, time>):

**A: [<12,20>, <5,15>, <2,20>, <15, 12>, <7, 25>, <8, 24>]**

**B: [<11,14>, <8, 16>, <3, 17>, <12, 12>, <10, 16>, <8,16>]**

**Exercise 1:** Get some graph paper and plot A and B (separately) with price on the x-axis and time on the y-axis.

**Exercise 2:** The giving orderings of A and B are not sorted according to the rules specified in the handout. Reorder each of A and B according to the rules (recall: order in non-decreasing order of price; options with identical price are ordered by time -- i.e., as a tie-breaker).

**Exercise 3A:** Neither A nor B are "pareto" -- i.e., they contain options that are dominated by other options (i.e., these dominated options are useless). Identify and prune the dominated options and duplicate options in both A and B. Retain the sorted ordering of the non-dominated options from Ex. 2. Now you have two pareto-sorted option lists. Let's call these A\_ps and B\_ps.

**Exercise 3B:** Repeat Ex. 3A, but this time try to see if you can devise systematic rules to identify dominated options as you examine the list from left-to-right (i.e., in a single pass).

Observation: since the lists are sorted, can the first option be dominated? NO! It has the minimum price and any other option with the same price must have equal or greater time (i.e., a duplicate in which case, only one of the copies is to be retained).

Now think about the 2nd option in the list. Under what conditions must it be pruned?

**Exercise 4A (union\_pareto\_sorted):**  Construct the sorted *union* of A\_ps and B\_ps (as in set-union; since A\_ps and B\_ps are pareto, they must be sets right?). Now prune dominated options in the union (if any).

**Exercise 4B:**  Now (if you didn't already in 4A), try to develop a systematic procedure for constructing the pruned-and-sorted union. Start by figuring out what must be the *first* option in the pruned-sorted union. After that?

**Exercise 5A (join\_plus\_plus):** Re-read the specifications of the **join\_plus\_plus** function. Now, enumerate all pairs of options between A\_ps and B\_ps and for each pair, apply the "plus-plus" rule to generate a new candidate option. This may be a little tedious, but just work through it. You might draw a matrix with the options from A\_ps associated with rows and the options from B\_ps associated with the columns; populate each matrix entry with the corresponding new candidate option.

**Exercise 5B:** Now sort and prune the options from 5A.

**Exercise 6 (join\_plus\_max):** Re-read the specifications for the **join\_plus\_max** function. Apply this operation to A\_ps and B\_ps. Recall, the resulting list will be sorted and pareto. Start by asking yourself,

"What must be the *first* option in the resulting list?"

It must have minimum price right? Only one pairing of options from A\_ps and B\_ps will give minimum price right? We claim that there is only one such possibility!

Now that you have identified the first option in the resulting list, think about the second. The 2nd option in the sorted-pareto resulting list will have higher price AND smaller time. An example to consider:

Suppose we have options <4,8> and <6,12> (from lists A and B respectively); joining these options gives <4+6, MAX(8, 12)> = <10, 12>. If you have more than $10 to spend, where should you spend it first? On the A-branch or the B-branch? How does this relate to the pareto-sorted lists for A and B?

**Exercise 7 (insert\_pareto\_sorted):**  Take A\_ps (the pareto-sorted option list you made in exercise 3) and consider a new option <4,13>. Is the new option dominated by a pre-existing option? Answer: no. Insert this new option according to the sorting rules. Are there pre-existing options that are now dominated by this new option? If so, eliminate them!