Shopping Plan—Google Code Jam Practice Problem

6/24/2009

I do not know the answer yet. For record, I document the process of thinking.

0. First Idea

For each item, there may be several stores providing it. For all the items, the combination of stores choices can cover all the cases. Suppose we have m items to purchase and there are n stores total. The worst situation is n^m cases.

Second step is to combine the items in same stores and get the total stores we need to go. This step may result worst case $min(m, n)$ stores. let's say, r stores are resulted.

Third step is to decide the best travelling path to get all the items, consider some perishable item may require you go home first before purchasing any more items in other store. suppose there are s special stores in which we purchase perishable items. we loop on all the possible paths (sequences of stores), there are $r!$ cases.

This is the easiest algorithm I can imagine, still I am not sure how to get the optimized travelling path.

1. Consider possible way to address it.

choose as less stores as possible

choose as cheaper items as possible

as less rounds as possible (if there are n perishable items, then at least n rounds are necessary.)

choose the best travelling path to cover all the stores

(*Hindsight: Those idea may be useful for the approximaiton algorithm, but can not ensure the accurate result.)

2. Focus on optimization of path for stores.

It should already be addressed in Graph Theory. or maybe it is **NP Complete**. If so, we can only resort to brute force and assuming the size of the problem is small.

(Actually, I get the correct answer for small input with brute force. It cost me about 8 hours programming and debugging. Considering that I have used the existing Java code to generate the combination of all path sequences, which I developed before when learing Volume 4 of *The Art of Computer Programming*, I actually spent more effort to addres this problem.)

3. The benefit of Literate Programming

It it the first time I experience the benefit of Literate Programming. It seems the process of documenting your rough idea can help you classify the idea and make it clear. Or just because you spend effort to say it out, you have gotten familiar with the problem domain and solution in your mind naturally and improve you ability and efficient to think about it.

The structure provided by CWEB is actually helpful for you to organize the code better, at least for the program to address the algorithm problem. The reason is that although we can use method to structure the algorithm in Java, the parameter used to transer information between method will cause inconvinient. but in CWEB, in program level you only have one method, there is no problem of information sharing and no extra redundant parameters, but still keep same abstract structure.

Next step is to refactor the java code to improve the maintainability.

4. System.out.printf

Good to System.out.printf, which is handy at format the number for output.