A. Habrard Master DSC/MLDM

Advanced Algorithms

Examination – 13th December 2022

Duration: 2 h Number of page(s): 2.

Only documents from the course or personal notes are permitted. Before you begin, it is recommended to read all the document.

Grading scale (temporary)

Part	1	2	3
on	5 pts	3 pts	12 pts

Part 1 Complexity (5 points)

1. Order increasingly the following functions, according to their growth rate $O(\cdot)$:
   $(\sqrt{2})^n$, $\sqrt{n^3}$, $\cos(n)$, $n\log(n)$, $n+\sqrt{n}$.

2. True or false? Justify your answer.
   If $f_1(n) \in O(f_2(n))$, $g(n) \in O(f_1(n))$ and $h(n) \in O(f_2(n))$ then $g(n)+h(n) \in O(f_2(n))$.

3. Consider the following two functions f_1 and f_2:
   $$f_1(n) = \begin{cases} n, \text{if } n \text{ is odd} \\ \sqrt{n}, \text{if } n \text{ is even and } n \leq 256 \\ n^3, \text{if } n \text{ is even and } n > 256 \end{cases}$$
   $$f_2(n) = \begin{cases} n \text{ , if } n \leq 250 \\ n^2, \text{if } 250 < n \leq 255 \\ n^3, \text{if } n > 255. \end{cases}$$
   Give the true assertion(s) among the following ones:

   (a) f_1 is $O(f_2)$,

   (b) f_2 is $O(f_1)$,

   (c) f_1 and f_2 are incomparable.

4. Solve the following recurrence:

   - $T(n) = 3T(n/2) + 2n^2$, for $n \geq 2$ and $T(1) = 1$
   - $T(n) = T(\sqrt{n}) + n$, for $n \geq 4$ and $T(2) = 1$.

Part 2 Branch and Bound (3 points)

We consider the Traveling Salesman Problem (TSP). Recall the problem and propose a branch and bound solution which does not have to store all possible candidate solutions in memory and makes use of information on the distances between the cities (presenting a solution similar to the one presented in the course slides is not accepted). You are expected to provide your solution as a pseudo-code. Explain also why it has a big chance to delete some branches in the search tree. Do not hesitate to give a graphical illustration of your solution if you think that it is appropriate.

Part 3 Traveling and taxes (12 points)

For his job, a salesman has to visit n cities $(c_1, \ldots, c_n)$ by car. Note that this problem is different from the TSP one seen in the previous section. The list of cities is fixed and the order of the visits is imposed by the numbering: c_1 is the starting point, c_2 the second city, and so on. The distance between 2 cities is expressed in kilometers and given by the quantity $d(c_i, c_j)$ for the distance between c_i and c_j. We assume that the car consumes 1 liter of gasoline per kilometer (for the sake of simplicity). The car has a capacity of C liters of gasoline and we assume that it is enough to reach to consecutive cities, *i.e.* $d(c_i, c_{i+1}) \leq C$ for $i = 1$ to $n-1$. We assume that the car is full of gasoline when the traveler starts his trip from c_1. When he reaches a city c_i, **he has two possibilities**: (1.) continue his trip or (2.) stop and refuel the car to the maximum capacity C. Note that the traveller has also to refuel completely the car at his final destination c_n.

1. The traveler wants to plan a trip with the minimum number of stops. Propose a greedy approach and prove that it is optimal. Give the complexity.
2. Now, the traveler wants to minimize the number of liters of gasoline added to the car. Does the previous approach provide an optimal solution? Justify.
3. Now, we assume that at each city, when the traveler decides to refuel then he has to pay a tax in euro corresponding the square of the number of liters of gasoline remaining in the car. This tax is also due when he reaches his final destination. For example, if the traveler reaches a city c_i with 4 liters of gasoline in the car, if he decides to refuel, he will pay a tax of 16 euros. The problem now is then to find a solution in order to pay the minimum amount of money.
   (a) Propose a greedy approach. Is it optimal?
   (b) We propose now to build a solution based on the dynamic programming paradigm. In order to help you in the construction of the solution, we propose to consider the following quantities (but you are free to use a different strategy if you prefer):
      - $paid(i, c)$: minimum amount of money paid for the trip from city c_i until final destination c_n when the traveler arrives at the city c_i with c liters of gasoline in the car. **Note that this is the quantity we want to optimize**.
      - $d(i, j)$: number of kilometers between point i and point j, in particular the quantity $d(i, i+1)$ will be of particular interest.
      - C: the capacity of the bottle. Above note that in $paid(i, c)$ we must have $c \leq C$.

      Questions:

      i. Give the structure of an **optimal solution** (and if possible prove its optimality).
      ii. Give a recursive definition of the optimal value *paid*. Pay attention to the limit and feasible cases.
      iii. Provide an algorithm for finding the solution in polynomial time. You must provide a complete pseudo-code for all solution. Give the complexity (time and space).
      iv. Provide an approach for computing the solution (*i.e.* the trip). Note that if you followed the setting above, the solution should be given by $paid(1, C)$.
      v. Apply your algorithm to the following example: the car has a capacity of 10 liters, and the distances between two consecutive cities are as follows: $d(c_1, c_2) = 1$, $d(c_2, c_3) = 7$, $d(c_3, c_4) = 2$, $d(c_4, c_5) = 6$, $d(c_5, c_6) = 3$, $d(c_6, c_7) = 5$. c_1 is of course the starting point and c_7 the final destination.