Algorithms, 2019/06/24

1. (20) Let c and d be 2 constants. Suppose that the time complexity of an algorithm is

$$T(n) = \begin{cases} c, & \text{if } n \leq 10 \\ T(\lfloor n/5 \rfloor) + T(\lfloor 7n/10 \rfloor) + dn, & \text{if } n > 10. \end{cases}$$

   Proof that the time complexity of the algorithm is linear.

2. (20) Let $X = x_1, x_2, \ldots, x_n$ be a sequence of n integers. A sub-sequence of X is a sequence obtained from X by deleting some of its elements. Design an $O(n^2)$ algorithm to find the longest monotonically increasing sub-sequence of the sequence X.

3. (20) Let v be a vertex in a graph $G = (V, E)$. The *eccentricity* of v is $f(v) = \max_{w \in V}\{d(v, w)\}$, where $d(v, w)$ is the number of edges on the path from v to w. The center of G is a vertex of minimum eccentricity. Design an efficient algorithm for finding the center of a tree, and determine its time complexity.

4. (20) Let n be a large composite integer.

   (a) Design a non-deterministic algorithm for factoring n.

   (b) Design a probabilistic algorithm for factoring n.

   (c) Assume that n is a product of two large distinct primes p and q, and $p, q \approx \sqrt{n}$. What is the probability of your algorithm in (b) to factor n?

5. (20) Let $x_1, x_2, \cdots, x_n$ be a data set of n integers. The *mode* of the n integers is the one that appears most often in the data set. We are going to design an algorithm for finding the mode of a very large data set by using very small amount of memory. In the following C-like code, valuables a and c are used to record some element a occurs c times.

```
c = 0;
for (i = 1; i ≤ n; i = i + 1) {
  read next x;      // read x_i
  if (c = 0) {a = x; c = 1;}
  elseif (x = a) {c = c + 1;}
  else {c = c − 1;}
}
print a;
```

   (a) Let the n input integers be $1, 2, 3, 3, 1, 3, 3$. Simulate the execution of the program, and show that the output is correct.

   (b) Explain why the above algorithm works for the case that the mode occurs at least $(n + 1)/2$ times in the input.

   (c) If no data occur at least $(n + 1)/2$ times, give an example to show that the algorithm will fail.