Algorithms, 2017/07/10

1. (20) Suppose you have designed an algorithm with time complexity

$$T(n) = O(tn^{1+1/t} + n \log n),$$

where n is the size of input and t is a parameter that you can choose between 1 and n. How to choose the parameter t so that your algorithm has the lowest time complexity?

2. (20) Fibonacci numbers are defined as: $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n > 1$. The value of F_n can be computed by the definition iteratively.

```
F_0 = 0; F_1 = 1;
for (i = 2; i ≤ n; i++)
  F_i = F_{i-1} + F_{i-2};
print F_n;
```

It can also be computed by the formula

$$F_n = (\theta^n - \bar{\theta}^n)/\sqrt{5},$$

where $\theta = (1 + \sqrt{5})/2$, and $\bar{\theta} = (1 - \sqrt{5})/2$. Compare the two methods for computing F_n. Which method is more efficient, i. e. which method needs less number of bit operations?

3. (20) Suppose you have reduced, in polynomial-time, a problem P to another problem Q. For each one of following facts, state what can be concluded by this reduction.

   (a) The problem Q is *NP*-hard.

   (b) The problem Q is in P.

   (c) The problem P is *NP*-hard.

   (d) The problem P is in P.

4. (20) Let $x_1, x_2, \cdots, x_n$ be a data set of n integers. The *mode* is the value that appears most often in a set of data. We are going to design an algorithm for finding the mode in 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 denote a occurs c times.

```
c = 0;
for (i = 1; i ≤ n; i = i + 1) {
  if (c == 0) {a = x_i; c = 1;}
  elseif (x_i == a) {c = c + 1;}
  else {c = c - 1;}
}
print a;
```

   (a) Explain why the above algorithm works for the case that some data occurs at least $(n + 1)/2$ times.

(b) If no data occur at least $(n+1)/2$ times, show that the algorithm will fail.

5. (20) We are going to design an approximation algorithm for the traveling salesman problem. The input to the problem is a weighted complete graph $G = (V, E, w)$ where w is a weight function $w : E \to R^+$ and R^+ is the set of positive numbers. Assume that w satisfies *triangle inequality*, i. e. $w(x, y) + w(y, z) \geq w(x, z)$ for all $x, y, z \in V$. The output of your algorithm should be a *good* solution C, which is a spanning cycle of G.

   (a) Let T be a spanning tree of G, and $v_1, v_2, \ldots, v_n$ be the depth-first traversal of T starting from some vertex v_1. Let $C = v_1, v_2, \ldots, v_n, v_1$ be a spanning cycle of G constructed from the depth-first traversal of the spanning tree T. Show that the total distance of C, $w(C)$, is bounded by $2w(T)$. (10)

   (b) Design an approximation algorithm for computing a spanning cycle C of G with $w(C) \leq 2w(C^*)$, where C^* is the optimal solution. (20)