## BBM402: Study Problems

1. Write the recursive relation for the running time of the algorithm given to solve ’Tower of Hanoi’ problem and provide a solution for it.

**Solution:**

$$T(n) = 2T(n-1) + 1.$$

The solution is $T(n) = 2^n - 1$. This can be shown by using induction on $n$. The proof by induction should consist of two steps: 1) base case, 2) inductive case, typically making use of the information called inductive hypothesis. The detailed solution is in the reading for ’Solving Recurrences’.

2. The running time for the mergesort algorithm is given as

$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + O(n).$$

Use a recursion tree, to show that $T(n) = O(n \log n)$. Show all your work.

**Solution:** (See class notes or the slides or the reading on ’Solving recurrences’ for the solution.)

3. Give an example to a divide-and-conquer algorithm (recursive) to multiply two n-digit numbers with $O(n^2)$ running time. Do you know any other algorithm that runs faster, meaning with $o(n^2)$ running time? If yes, describe that algorithm and its running time as well.

**Solution:** (See reading on ’Recursion’, Section 1.8)

4. Consider the recurrence relation $T(n) = 2T(n/2) + n/\log n$. Use a recursion tree, to show that $T(n) = \Theta(n \log\log n)$.

**Solution:** See the slides.

5. Consider the recurrence relation $T(n) = T(\sqrt{n}) + 1$. Use a recursion tree to show that $T(n) = \Theta(\log\log n)$. Similarly, if $T(n) = \sqrt{n}T(\sqrt{n}) + n$, show that $T(n) = \Theta(n\log\log n)$.

**Solution:** See the slides and reading on 'Solving Recurrences'.

6. - Karatsuba's multiplication algorithm has running time $T(n)$, where $T(n) = 3T(n/2) + n$. Use a recursion tree to show that $T(n) = \Theta(n^{\log_2 3})$.
   - Let $T(n) = 2T(n/2) + n/\log n$. Show that $T(n) = \Theta(n\log\log n)$.
   - Let $T(n) = 4T(n/2) + n/\log n$. Show that $T(n) = \Theta(n^2)$.

**Solution:** See the reading on 'Solving Recurrences'.

7. Q1: Argue briefly why the recursive relation for the running time of the Tower of Hanoi problem is T(n) = 2T(n-1)+1 for n at least 1, in an recursive algorithm provided in class. In other words, explain why we cannot do better with a shorter running time or why not more running time is needed in the given recursive algorithm.

**Solution:** See reading on recursion and class notes.

8. a) Write a short recursive program to calculate Fibonacci number n.
   b) In the recursion tree for this algorithm, give the number of leaves in this tree and prove that it is correct by induction.

**Solution:** See reading on dynamic programming.

9. In the conflict-free class-scheduling problem, a) prove that at least one maximal conflict-free schedule includes the class that finishes first.
   b) prove that the greedy algorithm that picks the non-conflicting class that finishes first from the available classes outputs an optimal schedule (with maximal number of classes).

**Solution:** See reading on greedy algorithms.

10. Design a Turing machine by writing the steps of the algorithm (no diagram asked) executed to accept if the input is in the language $A$ and reject otherwise.
    - $A = \{w\#w | w \in \{0,1\}^*\}$
    - $A = \{0^{2^n} | n \geq 0\}$
    - $A = \{0^n 1^n | n > 0\}$
    - $A = \{0^n 1^n 2^n | n > 0\}$

**Solution:** All of the questions are examples in Sipser's book (see webpage) Chapter 3 and our slides. Therefore, you can find the solutions there.

11. Besides these, I may give you a Turing machine with its diagram and an input, where you are supposed to list all configurations showing the executions of the Turing machine and the final state (accept or reject). (See Sipser's book, exercises 3.1 and 3.2)
12. Show that if the language $L$ and its complement $\bar{L}$ are recognizable, then $L$ is decidable. (Lecture Notes)
13. Show that $L_D$ is not recursively enumerable (recognizable). (Lecture Notes)
14. Show that $A_{TM}$ is not a decidable language. (pg.174 in Sipser's book/ Lecture Notes)
15. What is the 'Halting problem'? (pg.174 in Sipser's book)
16. Describe the (universal) TM that recognizes $A_{TM}$. (Lecture Notes)
17. Let $L$ be an undecidable language. Is it possible that both $L$ and $\bar{L}$ (the complement of $L$) are recognizable. If yes, give an example. If no, give a short proof. (pg.181 in Sipser's book)
18. What is the maximum independent set problem? What is the decision problem equivalent to this problem? (Lecture 7)
19. Show that the maximum independent set problem is in NP. (Lecture 7)
20. What is the minimum vertex cover problem? What is the decision problem equivalent to this problem? (Lecture 7)

21. Show that the minimum vertex cover problem is in NP. (Lecture 7)

22. Show that the independent set problem can be reduced to the clique problem in polynomial time? How big is this polynomial in the worst-case? (Lecture 7)

23. Show that the independent set problem can be reduced to the vertex cover problem in polynomial time? How big is this polynomial in the worst-case? (Lecture 7)