University of Nevada, Las Vegas Computer Science 477/677 Fall 2023

Study Guide for Examination October 25, 2023

1. Review answers to homework3:
   `http://web.cs.unlv.edu/larmore/Courses/CSC477/F23/Assignments/hw3ans.pdf`

2. Review answers to homework4:
   `http://web.cs.unlv.edu/larmore/Courses/CSC477/F23/Assignments/hw4ans.pdf`

3. Review answers to homework5:
   `http://web.cs.unlv.edu/larmore/Courses/CSC477/F23/Assignments/hw5ans.pdf`

4. Use Huffman's algorithm to find an optimal prefix-free binary code for the following weighted alphabet.

A	2
B	3
C	7
D	4
E	8
F	15

5. Solve each recurrence using the anti-derivative method.

   (i) $F(n) = F(n-2) + \frac{1}{n}$

   (ii) $F(n) = F(n - \sqrt{n}) + 1$

   (iii) $F(n) = F(n - \log n) + \log^2 n$

6. Solve each recurrence using the master theorem.

   `https://en.wikipedia.org/wiki/Master_theorem_(analysis_of_algorithms)`

   (iv) $F(n) = 2\sqrt{n} + logn$ (Use substitution.)

   (v) $F(n) = 3F(n/2) + 1$

   (vi) $F(n) = 4F(n/2) + n^2$

7. Solve each recurrence using the generalized master theorem, also known as the Akra-Bazzi method.

   `https://en.wikipedia.org/wiki/Akra%E2%80%93Bazzi_method`

   (vii) $F(n) = 3F(n/3) + 3F(2n/3) + n$

   (viii) $F(n) = 3F(n/3) + 3F(2n/3) + n^2$

   (ix) $F(n) = 3F(n/3) + 3F(2n/3) + n^3$

   (x) $F(n) = 3F(n/3) + 3F(2n/3) + n^4$

8. Consider the following recursive program for a function F.

```
int F(int n)
 {
  if(n <= 3) return n;
  else return (F(n/2)+F((n+1)/2)F((n+2)/2+F((n+3)/2)+n*n)
 }
```

We assume that each arithmetic operation takes constant time. The answer to each of the four questions belos is either $\Theta(\log n)$, $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^2)$, or $\Theta(n^2 \log n)$.

(a) What is the asymptotic complexity of $F(n)$? (Hint: Use the master theorem.)

Suppose you wish to find the value of $F(n)$ for some fixed positive integer n. Give asymptotic answers to the following questions.

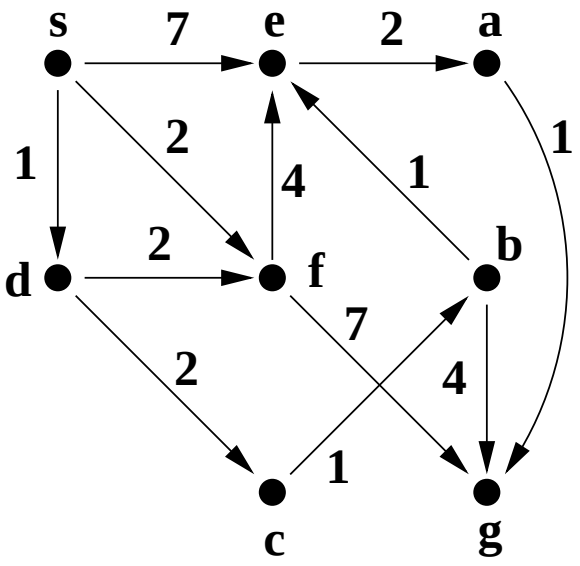
(b) What is the time complexity of your calculation if you use the recursive code given above? (Hint: Use the master theorem.)

(c) What is the time complexity of your calculation if you use dynamic programming?

(d) What is the time complexity of your calculation if you use memoization?[1] Note: The memos are stored in a search structure. Assume that it takes only constant time to find an item in that structure. That's actually false: the time to find a memo could be the logarithm of the number of memos, but just ignore that factor.
Memoization is explained at "`https://en.wikipedia.org/wiki/Memoization`"

9. Solve the recurrence: $F(n) = F(\log n) + 1$

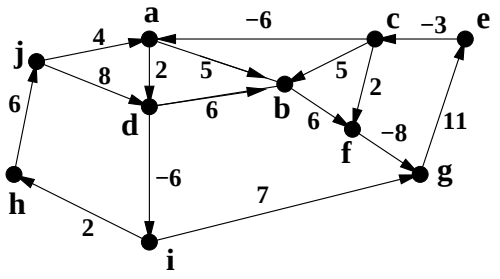
10. Walk through Dijkstra's algorithm for the single source minpath problem for the directed graph illustrated below, where **s** is the source vertex.



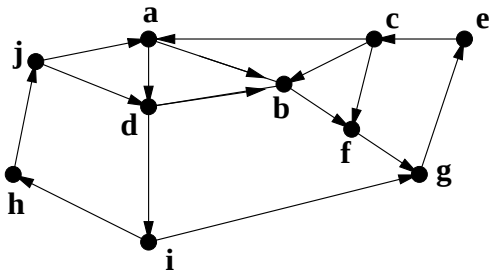
[1] We will cover memoization during the lecture on Monday October 23: if we don't get to it, it won't be on the exam.

11. Figure (a) below shows an instance of the all-pairs minpath problem. Work the first part of Johnson's algorithm on that graph, showing the adjusted weights in Figure (b).

    Do not complete the computation of Johnson's algorithm.



(a)



(b)

12. Compute the two square roots of $3 + 4\mathbf{i}$