

CS325 Midterm, Summer 2009, OSU

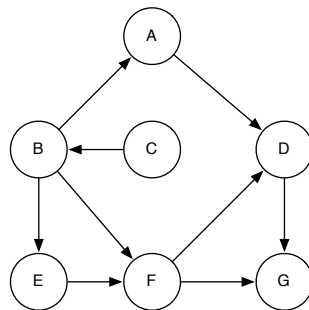
1 Rules

You are required to **type your solution**, and turn it in printed at the beginning of class, Thursday, 7/23.

You are forbidden to discuss this test in any way with any person other than the course instructor. This includes posting questions online. You are allowed to use any reference material or tools (i.e. text books, wolframalpha.com, calculators, compilers, etc.) to help you solve the problems.

2 Questions

- (5 points)** Prove the following facts:
 - $3n^4 + 5n^2 + 2 \in O(n^4)$
 - $\sqrt{n} \in O(n^2)$
 - $2^n \in O(4^n)$
 - $\log_2 n \in O(\log_5 n)$
 - For a complete graph $G = (V, E)$, $O(\log E) = O(\log V)$
- (3 points)** For each of the following algorithms, write a recurrence describing its runtime, and solve it (in big- O notation).
 - Divide an input of size n into 5 subproblems of size $n/2$, recursively solve the subproblems, then do $O(n)$ work combining the solutions.
 - Solve a problem of size n by recursively solving 2 subproblems, each with size $n - 1$, then do $O(1)$ work to combine.
 - Divide an input of size n into 9 subproblems of size $n/3$, recursively solve the subproblems, then do $O(n^2)$ work combining the solutions.
- (1 point)** Run topological sort on the following graph. Your answer should consist of a list of the vertices in a particular order.



4. **(2 point)** You are working for genetics company, and they want you to compress a data file containing a DNA sequence consisting of 10 million letters (the letters are A, C, T and G). Suppose the letter A appears 5 million times, C appears 2 million times, T appears 2.5 million times, and G appears 500,000 times. Give a prefix code for the letters A, C, T and G that uses the minimum average bits per letter. Show how you got the code.
5. **(2 point)** Suppose you have an inefficient implementation of a priority queue where the operations take the following amounts of time:

$$\text{insert} - O(n) \quad \text{extractMin} - O(n^2) \quad \text{decreaseKey} - O(n \log n)$$

- (a) What is the runtime of Dijkstra's algorithm using this data structure?
 (b) What is the runtime of Huffman's algorithm using this data structure?
6. **(3 points)** Consider the following recursive formula:

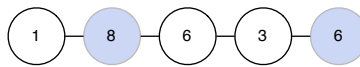
$$f(i, j) = \begin{cases} j & \text{if } i = 0 \\ i & \text{if } j = 0 \\ \min\{f(i-1, j) - f(i, j-1) + 3, 2f(i-1, j-1)\} & \text{otherwise} \end{cases}$$

Here i is in the range 0 to M , and j is in the range 0 to N .

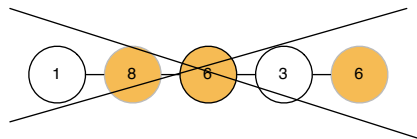
- (a) Write pseudocode for a dynamic programming algorithm to compute $f(m, n)$.
 (b) Analyze the runtime of your dynamic programming algorithm.
 (c) Write pseudocode for a memoized algorithm to compute $f(M, N)$.
7. A path is a special kind of graph $G = (V, E)$, with vertices v_1, v_2, \dots, v_n , and edges between v_i and v_j if and only if i and j differ by exactly 1. In this problem, each node v_i has a corresponding weight w_i (unlike previous problems, where weights were associated with edges). A subset of nodes is independent if no two of them are joined by an edge. The goal of this question is to solve the following problem:

In a graph G , which is a path, find an independent subset of nodes whose total weight is maximized.

For example, consider the following path:



The nodes that are shaded have a total weight of 14, which is the maximum of any independent subset. The next example is not a valid independent subset, because two nodes in the subset share an edge:



- (a) **(1 point)** Give an example that shows that the following greedy algorithm does not always find an independent set of maximum weight.

```

1   $S \leftarrow \{\}$ 
2  while some node remains in  $G$ :
3      Pick a node  $v_i$  of maximum weight
4       $S \leftarrow S \cup \{v_i\}$ 
5      Delete  $v_i$  and its neighbors from  $G$ 
6  return  $S$ 

```

- (b) (1 point) Give an example that shows that the following algorithm does not always find an independent set of maximum weight.

```

1  Let  $S_1$  be the set of all  $v_i$  where  $i$  is an odd number
2  Let  $S_2$  be the set of all  $v_i$  where  $i$  is an even number
3  Determine which of  $S_1$  or  $S_2$  has the greater total weight and return this one.

```

- (c) (3 points) Let $W[i]$ be the maximum total weight that can be obtained using the first i nodes in a path. Write a recursive formula for $W[i]$.
- (d) (1 point) Write pseudocode for a dynamic programming algorithm to calculate $W[n]$, where n is the number of nodes in a path.
- (e) (1 points) Analyze the runtime of your dynamic programming algorithm.
- (f) (3 points) Modify your algorithm so it returns the actual independent subset of nodes with the maximum total weight, rather than just the maximum total weight.

8. (3 points) Let G be an undirected graph with n vertices, where n is an even number. Prove the following theorem by contradiction:

If every vertex of G has degree at least $n/2$, then G is connected.

Recall that the degree of a vertex is the number of edges coming out of it, and a graph is connected if there is a path between all pairs of nodes. Hint: Think about connected components.

9. Consider the following algorithm for sorting a list of numbers (MERGE is the same function from mergesort):

```

1  TRISORT(A[1 ... n])
2      if  $n \leq 1$  return A
3       $L \leftarrow$  TRISORT( $A[1 \dots \frac{1}{3}n]$ )
4       $M \leftarrow$  TRISORT( $A[\frac{1}{3}n + 1 \dots \frac{2}{3}n]$ )
5       $R \leftarrow$  TRISORT( $A[\frac{2}{3}n + 1 \dots n]$ )
6       $LM \leftarrow$  MERGE( $L, M$ )
7       $LMR \leftarrow$  MERGE( $LM, R$ )
8      return  $LMR$ 

```

- (a) (2 points) What is the runtime of this algorithm? Explain your answer.
- (b) (3 points) Prove that TRISORT is correct by induction (you can assume MERGE is correct).

3 A note on writing psuedocode in LaTeX

I recently found a nicer way to format psuedocode in LaTeX than the CLRS style. The alternative is to use the alltt package (the .sty file is included with this template), which allows you to mix verbatim (i.e. monospaced) type, with math type. Formatting pseudocode with this package looks like this (random meaningless code):

```
allocate an array  $A[1..m, 1..n]$ 
for  $i = 0$  to  $n$ :
  for  $j = i$  to  $m$ :
     $X_{i,j} = i + j$ 
    if  $X_{i,j} = 7$ : return 3
```