COS 226 Algorithms and Data Structures Spring 2012

Final

This test has 14 questions worth a total of 100 points. You have 180 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet (8.5-by-11, both sides, in your own handwriting). No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Problem	Score
0	
1	
2	
3	
4	
5	
6	
Sub 1	

Problem	Score
7	
8	
9	
10	
11	
12	
13	
Sub 2	

Total	

Name:

Login ID:

Precept:

P01	Th 12:30	Diego Botero
P01A	Th 12:30	David Shue
P01B	Th 12:30	Joey Dodds
P02	Th 1:30	Josh Hug
P03	Th 3:30	Josh Hug
P04	F 11	Joey Dodds
P04A	F 11	Jacopo Cesareo

May 21: `14322d023dc5eda4`

0. **Initialization. (1 point)**

Write your name and Princeton NetID in the space provided on the front of the exam; circle your precept number; and write and sign the honor code.

1. **Analysis of algorithms. (10 points)**

(a) Suppose that you observe the following running times for an operation on a data structure with N items.

N	time
10,000	1.2 seconds
40,000	2.1 seconds
160,000	3.9 seconds
640,000	7.9 seconds
2,560,000	16.0 seconds

Estimate the running time of the operation (in seconds) as a function of N and use tilde notation to simplify your answer. Circle your answer.

(b) Consider the following data structure for representing a ternary search trie in Java.

```
public class TST<Value> {
    private Node root;                  // root of TST

    private class Node {                // inner class
        private char c;                 // character
        private Node left, mid, right;  // left, middle, and right subtries
        private Value val;              // value associated with string
    }
    ...
}
```

Suppose that you have a `TST` object with N key-value pairs, comprised of M `Node` objects. Using the 64-bit memory cost model from the textbook, how much memory (in bytes) does the object use as a function of M and N? Assume that the `Value` type is `Integer`—each `Integer` object uses 24 bytes. Simplify your answer using tilde notation.

(c) For each function on the left, give the best matching order of growth of the *running time* on the right. You may use an answer more than once or not at all.

__B__
```
public static int f1(int N) {
   int x = 0;
   for (int i = 0; i < N; i++)
       x++;
   return x;
}
```

_____
```
public static int f2(int N) {
   int x = 0;
   for (int i = 0; i < N; i++)
      for (int j = 0; j < i; j++)
         x += f1(j);
   return x;
}
```

_____
```
public static int f3(int N) {
   if (N == 0) return 1;
   int x = 0;
   for (int i = 0; i < N; i++)
      x += f3(N-1);
   return x;
}
```

_____
```
public static int f4(int N) {
   if (N == 0) return 0;
   return f4(N/2) + f1(N) + f1(N) + f1(N) + f4(N/2);
}
```

_____
```
public static int f6(int N) {
   if (N == 0) return 1;
   return f6(N-1) + f6(N-1) + f6(N-1);
}
```

_____
```
public static int f7(int N) {
   int x = 0;
   while (N > 0) {
      x++;
      N = N / 2;
   }
   return x;
}
```

A. $\log N$

B. N

C. $N \log N$

D. N^2

E. N^3

F. 2^N

G. 3^N

H. $N!$

2. **Miscellaneous. (9 points)**

Match each task on the left with the best-matching description on the right. For the purposes of this question, assume $P \neq NP$.

_____ Construct a DFA corresponding to a regular expression.

_____ Construct an NFA corresponding to a regular expression.

_____ Compress the strict majority of 10MB files by more than 3 bits.

_____ Find a longest path from s to t in an edge-weighted DAG.

_____ Find a longest path from s to t in an edge-weighted digraph.

_____ Compute a maximum s-t flow in a flow network using the Ford-Fulkerson algorithm with the shortest augmenting path rule.

_____ Compute a minimum s-t cut in a flow network given a maximum s-t flow.

_____ Determine whether a digraph is a DAG.

_____ Enumerate all subsets of a set of size N such that every pair of adjacent subsets differs in exactly one item.

_____ Find the longest key in an R-way trie that is a prefix of a query string.

A. Can be solved in linear time in the worst case.

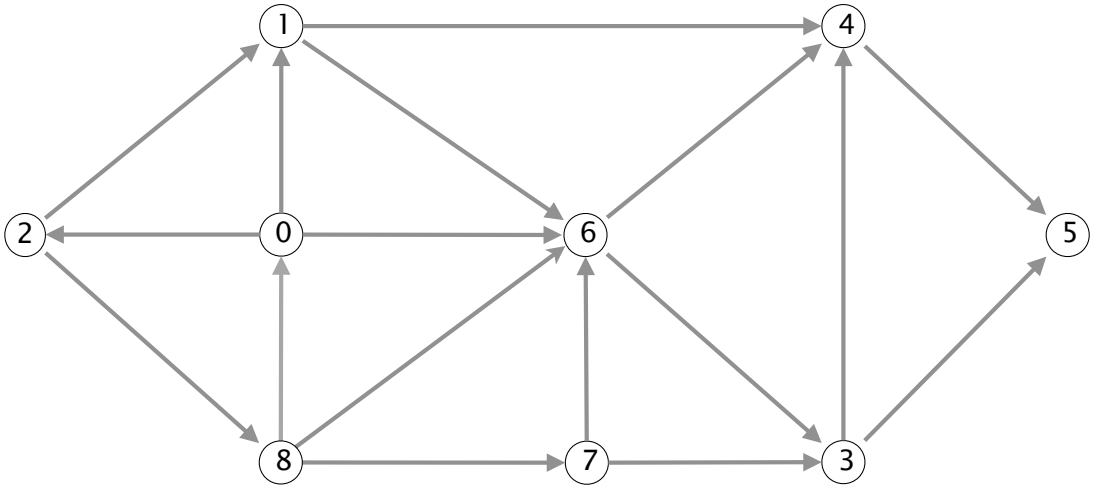
B. Can be solved in polynomial time in the worst case.

C. Can be solved in exponential time in the worst case.

D. Impossible.

3. **Graph search. (8 points)**

Consider the following digraph. Assume the adjacency lists are in sorted order: for example, when iterating through the edges pointing from 0, consider the edge $0 \to 1$ before $0 \to 2$ or $0 \to 6$.



(a) Run depth-first search on the digraph, starting from vertex 0. List the vertices in *reverse postorder*.

0

--- --- --- --- --- --- --- --- ---

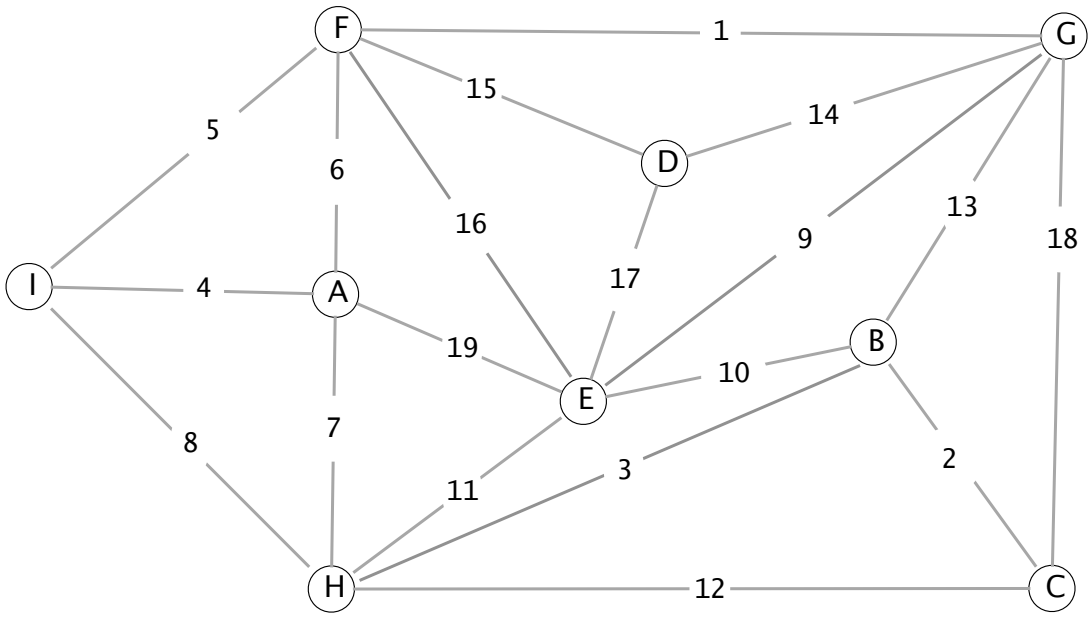
(b) Run breadth-first search on the digraph, starting from vertex 0. List the vertices in the order in which they are dequeued from the FIFO queue.

0

--- --- --- --- --- --- --- --- ---

4. **Minimum spanning trees. (8 points)**

Consider the following edge-weighted graph with 9 vertices and 19 edges. Note that the edge weights are distinct integers between 1 and 19.



(a) Complete the sequence of edges in the MST in the order that *Kruskal's algorithm* includes them (by specifying their edge weights).

1

---- ---- ---- ---- ---- ---- ---- ----

(b) Complete the sequence of edges in the MST in the order that *Prim's algorithm* includes them (by specifying their edge weights), starting from vertex A.

4

---- ---- ---- ---- ---- ---- ---- ----

5. **Shortest paths. (8 points)**

Suppose that you are running Dijkstra's algorithm on the edge-weighted digraph (below left), starting from a source vertex s. The table (below right) gives the `edgeTo[]` and `distTo[]` values immediately after vertex 7 has been deleted from the priority queue and relaxed.

edge	*weight*	*edge*	*weight*
$0 \to 2$	6.0	$5 \to 7$	10.0
$0 \to 4$	6.0	$5 \to 8$	4.0
$0 \to 5$	17.0	$6 \to 0$	12.0
$2 \to 7$	6.0	$6 \to 1$	3.0
$3 \to 0$	1.0	$6 \to 2$	1.0
$3 \to 1$	25.0	$6 \to 3$	17.0
$3 \to 6$	13.0	$6 \to 4$	9.0
$3 \to 8$	9.0	$6 \to 9$	4.0
$4 \to 5$	3.0	$6 \to 10$	4.0
$4 \to 6$	4.0	$7 \to 5$	11.0
$4 \to 7$	3.0	$7 \to 9$	6.0
$4 \to 8$	1.0	$7 \to 10$	7.0
$4 \to 9$	15.0	$8 \to 9$	13.0
$5 \to 1$	12.0	$9 \to 3$	4.0
$5 \to 2$	1.0	$10 \to 3$	5.0
$5 \to 4$	3.0	$10 \to 9$	2.0

v	`distTo[]`	`edgeTo[]`
0	∞	*null*
1	12.0	$5 \to 1$
2	1.0	$5 \to 2$
3	∞	*null*
4	3.0	$5 \to 4$
5	0.0	*null*
6	7.0	$4 \to 6$
7	6.0	$4 \to 7$
8	4.0	$5 \to 8$
9	12.0	$7 \to 9$
10	13.0	$7 \to 10$

(a) Give the order in which the first 5 vertices were deleted from the priority queue and relaxed.

				7

(b) Modify the table (above right) to show the values of the `edgeTo[]` and `distTo[]` arrays immediately after the next vertex has been deleted from the priority queue and relaxed. Circle those values that changed.

6. **String sorting. (6 points)**

Consider running LSD string sort on the input array `a[]` of 20 strings. Give the contents of the array `a[]` immediately after the fourth call to key-indexed counting for the indices 0–5 and 15–19.

i	a[i] (*input*)
0	heard
1	beard
2	cable
3	cache
4	scald
5	scale
6	medic
7	table
8	badge
9	weave
10	heave
11	hedge
12	fable
13	leave
14	wedge
15	ledge
16	media
17	rabid
18	sable
19	scare

i	a[i] (*fourth*)
0	
1	
2	
3	
4	
5	
6	badge
7	*not required*
8	*not required*
9	*not required*
10	*not required*
11	*not required*
12	*not required*
13	*not required*
14	leave
15	
16	
17	
18	
19	

7. **Substring search. (6 points)**

Below is a partially-completed Knuth-Morris-Pratt DFA for a string s of length 12 over the alphabet { `A` , `B`, `C` }. Reconstruct the DFA and s in the space below.

	0	1	2	3	4	5	6	7	8	9	10	11
`A`					5		7	1	9		11	
`B`					0		0	8	0		0	
`C`		0			0		0	4	0	10	0	
s					`A`		`A`	`B`	`A`	`C`	`A`	`C`

8. **Regular expressions. (6 points)**

Suppose that we run the RE-to-NFA construction algorithm from the lecture and textbook on the regular expression `( A ( B * | C ) * )`. The match transitions are shown below.

Circle which one or more of the following edges are in the ϵ-transition digraph.

$2 \to 5$	$2 \to 6$	$2 \to 7$	$2 \to 8$
$3 \to 4$	$3 \to 5$	$3 \to 6$	$3 \to 7$
$4 \to 3$	$4 \to 4$	$4 \to 5$	$4 \to 6$
$5 \to 6$	$5 \to 7$	$5 \to 8$	$5 \to 9$
$8 \to 2$	$8 \to 3$	$8 \to 7$	$8 \to 9$

9. **Ternary search tries. (6 points)**

Suppose that the following set of 11 strings are inserted into a TST (in some order).

```
CANNON    CAP    CHARTER    CLOISTER    COLONIAL    COTTAGE

IVY       QUAD   TERRACE    TIGER       TOWER
```

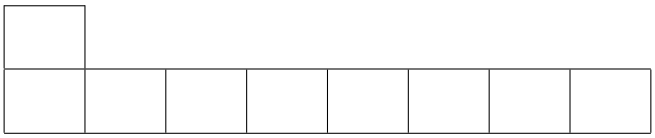
(a) Suppose that the strings are inserted into the TST in alphabetical order. What is the height of the resulting TST? (The height of a 1-node TST is 0.) Circle your answer.

(b) Suppose that the strings are inserted into the TST in an order which *minimizes* the height. What is the height of the resulting TST? Circle your answer.

10. **Burrows-Wheeler transform. (8 points)**

(a) What is the Burrows-Wheeler transform of the following?

```
A   B   C   C   C   B   C   B
```

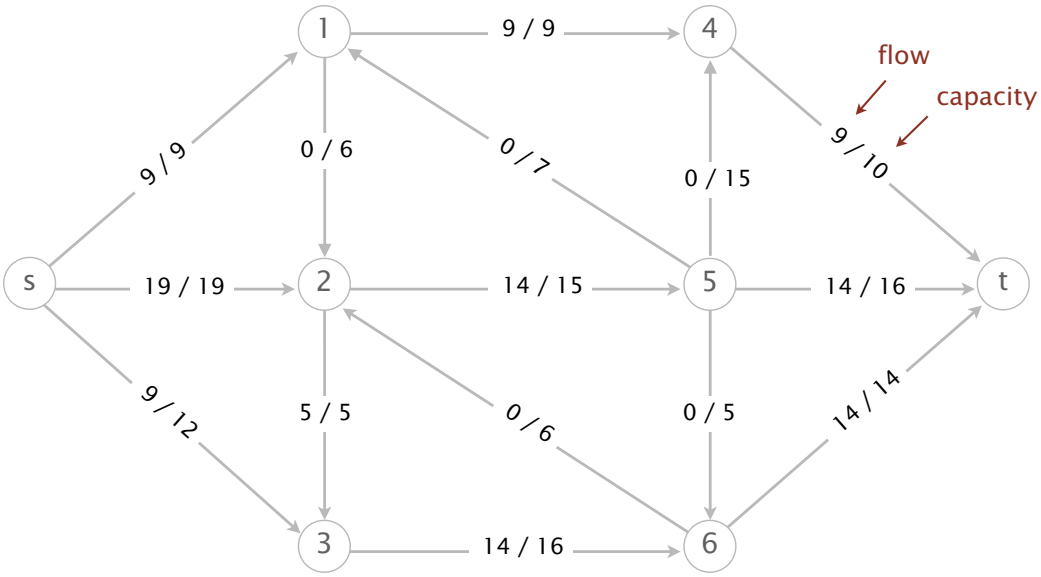


(b) What is the Burrows-Wheeler inverse transform of the following?

```
5
B   B   C   C   C   D   A   A   B
```

11. **Maximum flow. (8 points)**

Consider the following *st*-flow network and feasible flow f.



(a) What is the value of the flow f?

(b) Perform one iteration of the Ford-Fulkerson algorithm, starting from the flow f. Give the sequence of vertices on the augmenting path.

(c) What is the value of the maximum flow?

(d) List the vertices on the s side of the minimum cut.

(e) What is the capacity of the minimum cut?

12. **Algorithm design. (8 points)**

Given an edge-weighted digraph G the *bottleneck capacity of a path* is the minimum weight of an edge on the path.

For each part below, give a crisp and concise English description of your algorithm in the space provided. Your answer will be graded on correctness, efficiency, clarity, and conciseness.

(a) Given an edge-weighted digraph G, two distinguished vertices s and t, and a threshold value T, design an algorithm to find any one path from s to t of bottleneck capacity greater than or equal to T or report that no such path exists. The order of growth of the worst case running time of your algorithm should be $E + V$.

(b) Using the subroutine from (a), design an algorithm to find a *maximum bottleneck capacity path* from s to t in an edge-weighted digraph G. The order of growth of the worst case running time of your algorithm should be $(E + V) \log E$.

13. **Reductions. (8 points)**

- 3SUM. Given an integer array `a[]`, are there three indices (not necessarily distinct) `i`, `j`, and `k` such that `a[i] + a[j] + a[k] == 0` ?

- 3LINEAR. Given an integer array `b[]`, are there three indices (not necessarily distinct) `i`, `j`, and `k` such that `b[i] + b[j] = 8*b[k]` ?

(a) Show that 3LINEAR linear-time reduces to 3SUM. To demonstrate your reduction, give the 3SUM entries that would be constructed corresponding to the 3LINEAR instance:

b[]	b[0]	b[1]	b[2]	b[3]	b[4]

a[]

Hint: define M equal to 1 + maximum absolute value of any integer in `b[]`.

(b) Suppose that Alice discovers an $N^{1.9}$ algorithm for 3SUM and Bob discovers an $N^{1.5}$ lower bound for 3LINEAR. Which of the following can you infer from the fact that 3LINEAR linear-time reduces to 3SUM ?

I. There does not exist a $N^{1.4}$ algorithm for 3SUM.

II. 3SUM and 3LINEAR have the same asymptotic complexity.

III. There exists an $N^{1.9}$ algorithm for 3LINEAR.

(a) I only.

(b) I and II only.

(c) I and III only.

(d) I, II and III.

(e) None.