CS 373: Combinatorial Algorithms, Spring 2002
Midterm 1 — February 21, 2002

Name:
Net ID: Alias: U G

- This is a closed-book, closed-notes, open-brain exam. If you brought anything with you besides writing instruments and your handwritten $8\frac{1}{2}'' \times 11''$ cheat sheet, please leave it at the front of the classroom.

- Print your name, netid, and alias in the boxes above. Circle U if you are an undergrad, or G if you are a grad student. Print your name at the top of every page (in case the staple falls out!).

- You should answer all the questions on the exam.

- The last few pages of this booklet are blank. Use that for a scratch paper. Please let us know if you need more paper.

- If your cheat sheet if not hand written by yourself, or it is photocopied, please do not use it and leave it in front of the classroom.

- Submit your cheat sheet together with your exam. An exam without your cheat sheet attached to it will not be checked.

- If you are NOT using a cheat sheet you should indicate it in large friendly letters on this page.

- There are 20 multiple choice questions. There is only a single correct answer for each question. A correct answer would gain you one point, and incorrect answer gains you zero points (i.e., no penalty for incorrect answer). In case of several “correct” answers, you should chose the one which is better.

- Time limit: 75 minutes.

- Relax. Breathe. This is just a stupid midterm.

<table>
<thead>
<tr>
<th>#</th>
<th>Score</th>
<th>Grader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. What is the minimum depth of a network that can merge 1 item with \( n - 1 \) sorted items to produce a sorted sequence of \( n \) items?

A. 100  
B. \( \lg n \)  
C. \( n/2 \)  
D. None of the above

Answer: 

2. Which of the following is NOT an NP-hard problem?

A. Whether there exists a cycle that visits each vertex exactly once in a graph?  
B. Whether there exists a cycle that visits each edge exactly once in a graph?  
C. Whether there exists a tour in a graph that visits each vertex exactly once and has length greater than \( k \)?  
D. Whether a planar graph can be 3-Colored?

Answer: 

3. Consider the Greedy Minimum Set Cover algorithm, which repeatedly takes the largest set \( X \) in \( C \), and removes all elements of \( X \) from \( S \) and each subset in \( C \). Suppose we need to use at least \( k_{opt} \) subsets in \( C \) to cover \( S \). Which of the following statement about this algorithm is correct?

A. By this algorithm we can find \( k \) subsets in \( C \) to cover \( S \), where \( k \leq 2k_{opt} \).  
B. By this algorithm we cover at least \( n/2 \) elements from \( S \) after taking \( k_{opt} \) subsets.  
C. Neither (A) nor (B) is correct.

Answer: 

4. Statement: “All the problems in NP can be solved using polynomial space”. This statement is:

A. False.
B. True.
C. False only if \( P = NP \).
D. True only if \( P = NP \).
E. None of the above.

Answer: 

5. You are given a gate \( \text{Ord}(x, y) \) that receives as input two real numbers \( x, y \) and output true if \( x < y \) and false otherwise. Assume also that you can use or and and gates that have as many inputs as you want. You are required to build a circuit, with the smallest number of gates, and smallest depth that receives as input \( n \) numbers: \( x_1, \ldots, x_n \), and output true if \( x_1 < x_2 < \cdots < x_n \). The best possible circuit has:

A. \( n \) gates overall (of all types), and depth 2.
B. \( 2n \) gates overall (of all types), and depth \( 1 + \lfloor \lg n \rfloor \).
C. \( \Theta(n \log n) \) gates overall (of all types), and depth \( \Theta(\lg^2 n) \).
D. \( \Theta(n^2) \) gates overall (of all types), and depth \( \Theta(n) \).

Answer: 

6. **Problem: Interval Cover**

   **Instance:** A set \( I \) of \( n \) intervals \( I_i = [l_i, r_i], \ldots, [l_n, r_n] \), where \( l_i, r_i \) are integer numbers such that \( 1 \leq l_i, r_i \leq n \), and a parameter \( k \).

   **Question:** Are there \( k \) intervals \( I_{j_1}, \ldots, I_{j_k} \in I \) such that \( I_{j_1} \cup \ldots \cup I_{j_k} = [1, n] \).

This problem is:

A. NP-Complete.
B. NP-Hard.
C. Can be solved in polynomial time.
D. None of the above.

Answer: 

7. Consider these functions

\[ \log^*(\log n), \ (1 + \frac{1}{1000})^n, \ \log(\log^* n), \ \log(\log n) \log^* n, \ \log(\log n) \log n \]

What is the correct ordering of their asymptotic growth rate?

A. \( \log^*(\log n) \ll (1 + \frac{1}{1000})^n \ll \log(\log^* n) \ll \log(\log n) \log^* n \ll \log(\log n) \log n \)

B. \( (\log^* n) \ll \log(\log^* n) \ll (1 + \frac{1}{1000})^n \ll \log(\log n) \ll \log(\log n) \log^* n \ll \log(\log n) \log n \)

C. \( \log(\log^* n) \ll \log^*(\log n) \ll (\log^* n) \log^* n \ll (\log n) \log^* n \ll (1 + \frac{1}{1000})^n \ll (\log n) \log n \)

D. \( \log(\log^* n) \ll \log^*(\log n) \ll (\log^* n) \log^* n \ll (\log n) \log^* n \ll (\log n) \log^* n \ll (\log n) \log^* n \ll (1 + \frac{1}{1000})^n \)

E. \( \log(\log^* n) \ll \log^*(\log n) \ll (\log n) \log^* n \ll (\log^* n) \log^* n \ll (\log n) \log^* n \ll (\log n) \log n \ll (1 + \frac{1}{1000})^n \)

F. None of the above

Answer: 

8. The problem \textsc{Max-4SAT} is similar to \textsc{Max-3SAT}, but each clause has exactly 4 variables. Given an instance with \( m \) clauses of \textsc{Max-4SAT}, one can compute in polynomial time an assignment that:

A. Satisfies \((15/16)m\) of the clauses in expectation.

B. Satisfies \((7/8)m\) of the clauses in expectation. And furthermore, one can not do any better unless \( P = NP \).

C. The problem is NP-Hard and thus one can not find an assignment that satisfies more than \((1/8)m\) of the clauses in polynomial time.

Answer: 

9. The problem \textsc{2Coloring} (Deciding if a graph is colorable by two colors) is

A. NP-Complete.

B. Can be solved in polynomial time.

C. NP-Hard.

D. None of the above.

Answer: 

10. If we implement bitonic sort using hardware, then we can sort $n$ numbers in:

A. $O(\log n)$ time.
B. $O(\log^2 n)$ time.
C. $O(n)$ time.
D. $O(n \log n)$ time.
E. $O(n^2)$ time.

Answer: 

11. Given a graph $G$ with $n$ vertices, deciding if there is a path of length $\geq n/2$ which is simple (i.e., never visits a vertex more than once) is:

A. solvable in polynomial time.
B. NP-Complete.

Answer: 

12. Statement: “$NP$ is the set of all decision problems $X$, such that one can verify a positive answer to a given instance in linear time.”

This statement is:

A. False.
B. True.

Answer: 

13. One can do a 2-approximation to the TSP with triangle inequality for a graph $G$, because:

A. The triangle inequality implies that all edges of $G$ have the same weight up to a factor of 2.
B. The optimal TSP cycle $C$ in $G$ is a spanning graph of $G$, and the weight of $C$ is larger than the weight of the minimum spanning tree of $G$.
C. The TSP cycle in this case is just the minimum spanning tree of $G$.

Answer: 

14. The solution for the recurrence:

\[ A(n) = A(\lfloor \log n \rfloor) + 1 \]

is:

A. \( \Theta(n \log n) \)
B. \( \Theta(n^{\log n}) \)
C. \( \Theta(\log n) \)
D. \( \Theta(\log^* n) \)

Answer: [ ]

15. The problem of finding the largest independent set of vertices in a tree with \( n \) vertices is

A. NP-Hard.
B. NP-Complete.
C. Can be solved in \( O(n) \) time and this is the fastest algorithm possible.
D. Can be solved in \( O(n^2) \) time and this is the fastest algorithm possible.

Answer: [ ]

16. Given \( n \) bits \( b_1, \ldots, b_n \), one can build a circuit (using only standard AND/OR gates with two inputs) that decides if \( \sum_i b_i \geq n/2 \) using

A. \( O(n) \) AND/OR gates and of depth \( O(\sqrt{\log n}) \) and this is the best possible.
B. \( \Theta(n^2) \) AND/OR gates and depth \( O(n) \) and this is the best possible.
C. \( O(n \log^2 n) \) AND/OR gates and of depth \( O(\log^2 n) \) and this is the best possible.
D. This problem can not be solved using only AND/OR gates.

Answer: [ ]
17. What is the expected running time of $\textsc{Chance}(n)$?

```
\textsc{Chance}(n):
i \leftarrow \text{Random}(1, \ldots, n)
\ j \leftarrow \text{Random}(1, \ldots, n)
\text{if } i = j
\quad \text{halt;}
\text{else}
\quad \textsc{Chance}(n-1);
```

A. $\Theta(n)$
B. $\Theta(\lg n)$
C. $\Theta(n^2 \lg n)$
D. $\Theta(2^n)$

Answer: [ ]
18. Professor Kipod Metoral had declared that if you have $n$ horses in a room, then they must all be of the same color. This declaration had been met with disbelief in the horse raising community (which know that all their horses have different colors), and Professor Kipod Metoral had been forced to provide a proof of the claim. Here is the proof:

**Proof:** The proof is by induction. For the case $n = 1$ we have a single horse in the room, and it has a single color, so the claim is true.

Next, assume that we have $n + 1$ horses in the room for $n > 0$: $h_1, \ldots, h_{n+1}$. We remove one horse $h_1$ from the room. We are left with $n$ horses in the room (i.e. $h_2, \ldots, h_{n+1}$), and by induction hypothesis we know that they all have the same color. We now put $h_1$ back in the room, and remove the horse $h_{n+1}$. Again, we remain with $n$ horses (i.e., $h_1, \ldots, h_n$) and they all have the same colors. It follows that the color of $h_1$ is identical to the color of $h_2, \ldots, h_n$, and the color of $h_{n+1}$ is identical to the color of $h_2, \ldots, h_n$, and it thus follows that all the horses have the same color. QED.

This proof is incorrect because:

A. Horses are not a mathematical entity, and as such you can not argue about their colors.

B. The induction proof go in the wrong direction, it is going backward, but it should be going forward.

C. The base of the induction is too low, as the induction step fails for $n = 1$.

D. The proof is incorrect because it is using circular reasoning.

**Answer:**
19. Professor Sos Meshuga had declared that the following problem is NP-Complete:

**Problem:** HALTING

| Instance: A valid C program PROG (the program does not read any input). |
| Question: If we execute the program PROG, does PROG stop? |

And here is her proof:

**Proof:** We reduce 3SAT to HALTING, as follows: Given a formula $F$ over $n$ variables $x_1, \ldots, x_n$, generate a program $PROG_F$ that checks all the possible $2^n$ assignments to $F$, and for each such assignment decides if it is satisfying. If the program finds a satisfying assignment it stops. Otherwise, if all possible assignments to $F$ are not satisfying, the program gets into an infinite loop.

Clearly, $PROG_F$ has polynomial length in the length of $F$, and we can generate it in polynomial time. Furthermore if $PROG_F$ stops then $F$ is satisfiable, and if $PROG_F$ does not stop than $F$ is not satisfiable. Thus, we reduced 3SAT to HALTING in polynomial time and HALTING is NP-Complete. □

A. The proof is correct. HALTING is NP-Complete.
B. The proof is incomplete. HALTING is only NP-Hard.
C. The reduction is incorrect. HALTING is not even NP-Hard.
D. The proof is incorrect, as the reduction is in the wrong direction.

**Answer:** □
20. You are given an NP-Hard minimization problem PROBLEM (PROBLEM can be for example GENERAL TSP, or MAX 3SAT), and for any instance \( I \), the solution of PROBLEM on \( I \) is either exactly \( \mu \) or \( > c \cdot \mu \), where \( c > 1 \) is a constant, and \( \mu \) is a parameter that depends on \( I \). Then, there is no \( c \)-approximation algorithm for PROBLEM because:

A. If a problem is NP-Hard then it cannot be approximated to within any factor larger or equal to \( c \).
B. Such an algorithm can be used to solve the problem PROBLEM in polynomial time, which would imply that \( P = NP \).
C. Such an algorithm would imply that we cannot approximate SATISFIABILITY to within any constant factor.
D. This would imply that a woodchuck can chuck at most \( O(n) \) pieces of wood in constant time.

Answer: 

— The End —