Foundations of Computing

31 May 2010

Thore Husfeldt, ITU

Instructions

What to bring. You can bring any written aid you want. This includes the course book and a dictionary. In fact, these two things are the only aids that make sense, so I recommend you bring them and only them. But if you want to bring other books, notes, print-out of code, old exams, or today's newspaper you can do so. (It won't help.)

You can't bring electronic aids (such as a laptop) or communication devices (such as a mobile phone). If you really want, you can bring an old-fashioned pocket calculator (not one that solves recurrence relations), but I can't see how that would be of any use to you.

What do check. In the multiple-choice questions, there is one and only one correct answer. However, to demonstrate partial knowledge, you are allowed to check 2 or more boxes, but this earns you less than full points for that question. If the correct answer is not among your checked boxes, the score will even be negative. If you don't check anything (or check *all* boxes) your score is 0. Also, if you check boxes at random, your expected score is 0. For more details, read [Gudmund Skovbjerg Frandsen, Michael I. Schwartzbach: A singular choice for multiple choice. SIGCSE Bulletin 38(4): 34–38 (2006)].

1 Discrete maths only

1. Growth of functions.

(a) (*1 pt.*) Which pair of functions has the property that f is $O(g)$ and g is $O(f)$?

- [✔] $(x^4+1)(x+x^3)$ and x^7
- [B] $(x\log_2 x)(x+7)$ and x^2
- [C] $8x^2$ and $x!$
- [D] 2^x and 4^x

(b) (*1 pt.*) Which pair of values, C and k, are witnesses that show that $3x+26$ is $O(x)$?

- [A] $C=3, k=26$
- [B] $C=3, k=1000$
- [✔] $C=4, k=27$
- [D] $C=5, k=3$

(c) (*1 pt.*) Suppose f satisfies

$$f(n)=\begin{cases} n, & n \text{ is even,}\\ n+1, & n \text{ is odd.}\end{cases}$$

Which of the following statements is true?

- [✔] f is a function from $\mathbf{Z}$ to $\mathbf{Z}$
- [B] f is not a function because $f(5)=f(6)$
- [C] f is one-to-one
- [D] $f\circ f \neq f$

(d) (*1 pt.*) Assume that x, y, z are real numbers. Which of the following is the negation of the statement "x, y, and z are all positive".

- [A] x, y, and z are all negative.
- [B] $x\le 0$ and $y\le 0$ and $z\le 0$.
- [C] $x\ge 0$ and $y\ge 0$, but $z\le 0$.
- [✔] $x\le 0$ or $y\le 0$ or $z\le 0$.

(e) (*1 pt.*) Let S be the set of bit strings (strings of 0s and 1s) of length at least 2. Which of the following functions is *onto* S?

- [A] $f(s)=$ the string s with a 0 in front. (For example, $f(1)=01$.)
- [✔] $f(s)=$ the reversal of s. (For example, $f(011)=110$.)
- [C] $f(s)=$ the string obtained by changing all 0s in s to 1s. (For example, $f(01)=11$.)
- [D] $f(s)=$ the string s with its first character replaced by 0. (For example, $f(110)=010$.)

(f) (*1 pt.*) Define the set R of strings over $\{a, b, \ldots, z\}$ recursively as follows: (1) All one-letter strings consisting of a single vowel belong to R. (2) If the string α belongs to R then both zα and xα belong to R. Which of the following is a string in R?

- [A] zzzz
- [B] aeiou
- [✔] zxe
- [D] x

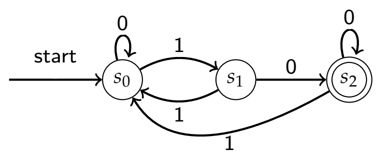
(g) (*1 pt.*) Let $f(n) = 2f(n/2) + 27n\log^2 n$ and $f(1) = 1$. Then, (give only the smallest correct answer)

- ✓ A $T(n) = O(n\log^3 n)$
- B $T(n) = O(n\log^2 n)$
- C $T(n) = O(n^{\log n}\log n)$
- D $T(n) = O(n^{\log_{27} n})$

(h) (*1 pt.*) If $S \subseteq T$ but $S \neq T$ then

- A Either S or T is empty
- ✓ B $T - S \neq \emptyset$
- C $S \cap \emptyset = T$
- D $S \cup T = S$

(i) (*1 pt.*) Which language is recognised by this deterministic finite-state automaton:



- A Bit strings ending in 10.
- ✓ B Bit strings that contain an odd number of 1s and end in 0.
- C Bit strings that contain at least two 0s.
- D Bit strings consisting only of 1s followed by a single 0.

(j) (*1 pt.*) Which strings are in the regular set $\mathbf{1}^*\mathbf{01}^*\mathbf{0}(\mathbf{0}\cup\mathbf{1})^*$.

- A Bit strings ending in 10.
- B Bit strings that contain an odd number of 1s and end in 0.
- ✓ C Bit strings that contain at least two 0s.
- D Bit strings consisting only of 1s followed by a single 0.

(k) (*1 pt.*) License plates on the planet Maximegalon consist of 3 or 4 alphabetic letters followed by 3 different digits, like this: XMA-321 or LAAX-042. The Maximegalon alphabet has only the letters A E G I L M N O X, and 6 different digits. How many different license plates are there?

- A $(\binom{9}{3} + \binom{9}{4})6^3$.
- ✓ B $(9^4 + 9^3)(6 \cdot 5 \cdot 4)$.
- C $\binom{9}{3}\binom{6}{3} + \binom{9}{4}\binom{6}{3}$.
- D $(9 \cdot 8 \cdot 7(6+1))(6 \cdot 5 \cdot 4)$.

(l) (*1 pt.*) Suppose A and B are events with $p(A) = \frac{3}{4}$ and $p(B) = \frac{1}{3}$. Which statement is true?

- A If A and B are independent then $p(A \cap B) = \frac{3}{7}$.
- ✓ B $p(A \cap B) \geq \frac{1}{12}$.
- C A and B cannot be independent.
- D A implies B.

2. The social network site *HateBook* allows people to express their hatred of others. Alice can declare that she *hates* Bob. Bob can hate her back, or not. If Bob hates her back, Alice and Bob are *enemies*. Moreover, the enemy of your enemy is your friend; this is the only way to make friends on HateBook. All HateBook users are well-adjusted individuals with good social skills, so they do not hate themselves. These assumptions naturally define binary relations H, E, and F on the set S of individuals.

Answer the following questions on a separate piece of paper. Be brief but precise, your correct use of mathematical notation and formalism is an important aspect of this question.

(a) (*1 pt.*) Construct a small HateBook network with three individuals that shows both hatred, enmity, and friendship. Describe it both as in set-theoretic terms, as a graph, and as a matrix.

(b) (*1 pt.*) List the properties (reflexive, symmetric, transitive, etc.) that the relation H has and doesn't have.

(c) (*1 pt.*) Can you be your own friend? If yes, give an example. If no, explain (one sentence) why this is so.

(d) (*1 pt.*) Is everybody always their own friend? If yes, explain (one sentence) why this is so. In no, give a counterexample.

(e) (*1 pt.*) Can somebody be both your enemy and your friend? If yes, give the smallest example. If no, explain (one sentence) why this is so.

2 Algorithms and data structures only

3.

(a) (*1 pt.*) Find the running time of the following piece of code:

```
for i = 1 to n
    insert (i^2, S)
for i = 1 to n
    remove (S)
```

Assume both **insert** (i, S) and **remove**(S) take time $O(\log |S|)$, and choose the smallest correct estimate.

- [A ✔] $O(n \log n)$
- [B] $O(n \log^2 n)$
- [C] $O(n^2)$
- [D] $O(n^2 \log n)$

(b) (*1 pt.*) Find the running time of the following piece of code:

```
int f(int n) {
    if n > 3 then { return f(n − 1) · f(n − 3); }
    else return 3;
}
```

(Choose the smallest correct estimate.)

- [A ✔] $T(n) = T(n-1) + T(n-3) + O(1)$
- [B] $T(n) = T(n-1) \cdot T(n-3) + O(1)$
- [C] $T(n) = 2T(n-1) + O(n-3)$
- [D] $T(n) = T(n-1) + O(n)$

(c) (*1 pt.*) Assume you have a data structure that maintains a set S under insertion. The operation "**insert**(s, S)" takes time $O(|S|)$ if $s \notin S$ and constant time if $s \in S$. After the operation, $s \in S$ holds.
Find the running time of the following piece of code, starting with an empty set S.

```
for i = 1 to n
    for j = 1 to n
        insert (j, S)
```

- [A] $O(n^{3/2})$
- [B ✔] $O(n^2)$
- [C] $O(n^2 \log n)$
- [D] $O(n^3)$

(d) (*1 pt.*) Assume I want the code from the previous exercise to run in total time $O(n \log n)$. What bounds on the running time of **insert** guarantee this result?

- [A] **insert** must take time $O(\log |S|)$
- [B] **insert** must take constant time
- [C] **insert**(s, S) must take $(\log |S|)$ time if $s \notin S$, and constant time otherwise
- [D ✔] This is impossible, no matter what you do.

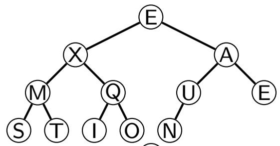
(e) (*1 pt.*) Consider a stack. A number i means "insert i" and "$*$" means "pop".

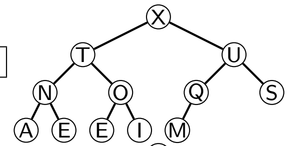
E * X A * * M Q * U E S * T * I O N

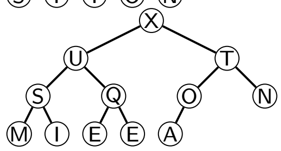
Give the sequence of values popped by these operations.

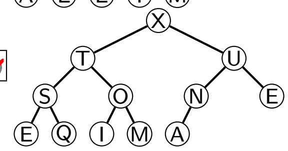
- [A] E X A M Q U E S T I O N
- [B] N O I T S E U Q M A X E
- [C ✔] E A X Q S T
- [D] E A X Q M S E U T N O I

(f) (*1 pt.*) Assume keys E X A M Q U E S T I O N are inserted in that order into an initially empty heap-based priority queue (prog. 9.5 in Sedgewick), using the `insert` method. What is the resulting data structure?

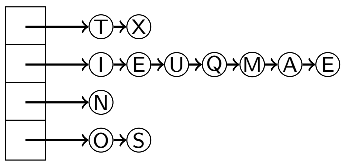
A 

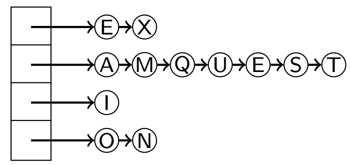
B 

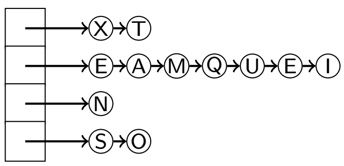
C 

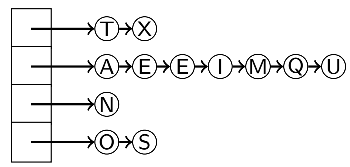
D ✔ 

(g) (*1 pt.*) Give the contents of a hash table where the keys E X A M Q U E S T I O N are inserted in that order into an initially empty table of size $M = 4$ using separate chaining. Use the hash value $k \bmod M$ to transform the kth letter of the alphabet (A = 1) into a table index 0, 1, 2, 3.

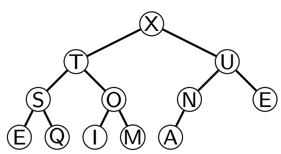
A ✔ 

B 

C 

D 

(h) (*1 pt.*) Give the inorder traversal of this tree:



A E X A M Q U E S T I O N

B ✔ E S Q T I O M X A N U E

C A E E I M N O Q S T U X

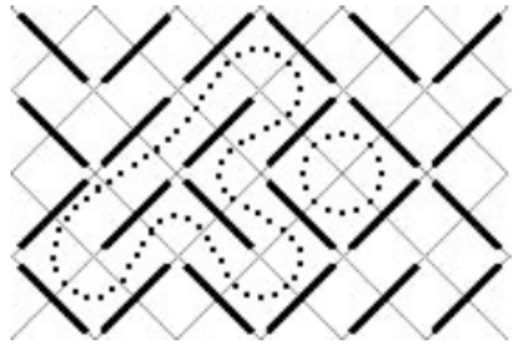
D E S Q T O I M X U N A E

4. This exercise asks you to suggest a solution to Slash Maze, which you can find on the next page. Don't write a full programme! Write your answer on a separate piece of paper; use English or Danish; be brief and precise (do not use more than one page, preferably less). One concrete, complete, well-chosen example is often the best way of explaining yourself. If you want, you can use some form of code or pseudo-code, but you don't have to.

(a) (*5 pt.*) Explain how you would solve this problem in an efficient way. (Polynomial time will be fine.) Do you use a data structure or an algorithm? Which ones? What is the resulting running time and space? Express yourself in terms of the parameters of the problem (e.g., w, h, k, l), instead of just saying things like "runs in $O(n)$" without telling me what n is.

Slash Maze

By filling a rectangle with slashes (/) and backslashes (\), you can generate nice little mazes. An example is to the right. As you can see, paths in the maze cannot branch, so the whole maze contains only (1) cyclic paths and (2) paths entering somewhere and leaving somewhere else. We are only interested in the cycles. There are exactly two of them in our example. Your task is to write a program that counts the cycles and finds the length of the longest one. The length is defined as the number of small squares the cycle consists of (the ones bordered by gray lines in the picture). In this example, the long cycle has length 16 and the short one length 4.



Input

Each description begins with one line containing two integers w and h ($1 \le w, h \le 75$), representing the width and the height of the maze. The next h lines describe the maze itself and contain w characters each; all of these characters will be either "/" or "\".

Output

Output the line "*k* Cycles; longest has length *l*.", where *k* is the number of cycles in the maze and *l* the length of the longest of the cycles. If the maze is acyclic, output "There are no cycles." .

Example 1	Example 2

Sample Input
`6 4`
`\//\\/`
`\///\/`
`//\\/\`
`\/\///`
Sample output
`2 Cycles; longest has length 16.`
	Sample Input
	`3 3`
	`///`
	`\//`
	`\\\`
	Sample output
	`There are no cycles.`