[id=GenCS2012SpringQuizzes,hwexamtype=Quiz]

Quizzes for General CS II (320201) Spring 2012

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

April 8, 2013

Contents

Assignment 1: Graph Theory(Given Feb. 13. 2012)

Problem 1.1 (Properties of trees)

4pt

Please name three properties of trees (in the computer science sense of the word).

Solution: Any three of the following:

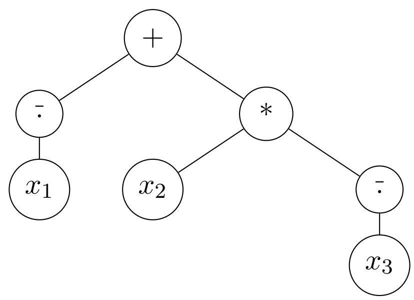
- DAG
- exactly one initial node (the root)
- all nodes but the root have in-degree 1
- for any node except the root, there is exactly one path from the root to it

---

Problem 1.2 (Parse Tree) 8pt

Draw the parse tree of the expression $\overline{x_1}+x_2 * \overline{x_3}$ and provide a mathematical representation of it.

Solution:



$G := \langle V, E, f \rangle$
$V := \{A, B, C, D, 1, 2, 3\}$
$E := \{\langle A, B \rangle, \langle A, C \rangle, \langle B, 1 \rangle, \langle C, 2 \rangle, \langle C, D \rangle, \langle D, 3 \rangle\}$
$f(A) = +,\ f(B) = f(H) = \bar{\cdot},\ f(C) = *,\ f(D) = x_1,\ f(F) = x_2,\ f(I) = x_3$

Assignment 2: Positional numbering systems(Given Feb. 20. 2012)

3pt

Problem 2.1 (Base conversions)
Convert the following numbers from the original base to the destination base, as specified:

1. 110101011110 from base 2 to base 16
2. 110111010111 from base 2 to base 8
3. $A39EF2$ from base 16 to base 2

Solution:

1. $110101011110_2 = D5E_{16}$
2. $110111010111_2 = 6727_8$
3. $A39EF_{16} = 10100011100111101111_2$

Problem 2.2 (Half adder) 8pt

1. Explain in 1-2 sentences what the Half adder is.
2. Draw the circuit of the Half adder.
3. Write down the cost and the depth of the Half adder.

Solution:

1. The half adder is a circuit that implements the addition of two 1-bit binary inputs by outputing their sum and a carry bit.
2. Please see slides.
3. Cost 2, depth 1.

Assignment 3: Number Systems(Given Feb. 27. 2012)

9pt

Problem 3.1 (Binary Number System Definitions)

- Given an $n+1$-bit binary number $a = a_n a_{n-1} \ldots a_0$, please provide a mathematical definition of the following number representations, i.e. provide a formula for calculating what decimal number they represent:
    1. $\langle\langle a \rangle\rangle$
    2. $(\langle\langle a \rangle\rangle^-)$
    3. $\langle\langle a \rangle\rangle^{2s}_{n+1}$
- Compute each of these three numbers for $a = 11100101$

Solution:

- We just have the definitions from class:
    1. $\langle\langle a \rangle\rangle = \sum_{i=0}^{n} a_i \cdot 2^i$
    2. if $a_n = 0$, $(\langle\langle a \rangle\rangle^-) = \sum_{i=0}^{n-1} a_i \cdot 2^i$
       if $a_n = 1$, $(\langle\langle a \rangle\rangle^-) = -\sum_{i=0}^{n-1} a_i \cdot 2^i$
    3. $\langle\langle a \rangle\rangle^{2s}_{n+1} = -a_n \cdot 2^n + \sum_{i=0}^{n-1} a_i \cdot 2^i$
- And after converting the numbers we get:
    1. $\langle\langle 11100101 \rangle\rangle = 2^7 + 2^6 + 2^5 + 2^2 + 2^0 = 229$
    2. $(\langle\langle 11100101 \rangle\rangle^-) = -\langle\langle 1100101 \rangle\rangle = -(2^6 + 2^5 + 2^2 + 2^0) = -101$
    3. $\langle\langle 11100101 \rangle\rangle^{2s}_{n+1} = \langle\langle 100101 \rangle\rangle^{2s}_{n+1} = -2^5 + 2^2 + 2^0 = -27$

Problem 3.2 (Simple Assembler) 3pt

Imagine that the data store is already full with values. Your task is to store in cell $P(7)$ the sum of the values in cells $P(0)$ and $P(7)$, incremented by 1.

Solution:

```
LOAD 0
ADD 7
ADDI 1
STORE 7
```

Assignment 4: Simple While(Given Mar. 05. 2012)

12pt

Problem 4.1 (Simple While)
A "star recurrence" has the following form:

$$S_n = 6 * S_{n-1} * (S_{n-1} - 1) + 1$$

Write a Simple While program (in both concrete and abstract syntax) that computes the n-th "star number" (computed with the recurrence above), given S_0.

Note: You can use any value for n and S_0, so, if you picked for example $n = 10$ and $S_0 = 6$, your concrete syntax program might start with:

```
var n:=10; var S0:=6;
```

Solution: Concrete syntax:

```
var n:=10; var S0:=6;
while ( n>0 ) do {
        S0 := 6 * S0 * (S0−1) + 1;
        n := n−1;
}
return S0;
```

Abstract syntax:

```
( [("n", Con 10), ("S0", Con 6)],
  While ( Leq(Con 0, Var "n"), Seq [
                  Assign("S0", Add(Mul(Con 6, Mul(Var "S0", Sub(Var "S0", Con 1))), 1)),
                  Assign("n", Sub(Var "n", Con 1))
                  ]),
                  Var "S0")
```

Assignment 5: VM Static Procedures(Given Mar. 12. 2012)

Problem 5.1 (The Tak function) 12pt

Write down a static $\mathcal{L}(\mathtt{VM})$ procedure that computes the value of the Tak function:

$$tak(x,y,z) = \begin{cases} z & \text{if } x > y \\ tak(tak(x-1,y,z), tak(y-1,z,x), tak(z-1,x,y)) & \text{if } x \leq y \end{cases}$$

Write short comments (e.g. in pseudocode like in the slides) indicating the major steps in your code and what they do.

Solution:

`proc 3 49`	$tak(x,y,z)$
`arg 2 arg 1 leq cjp 5`	if $x \leq y$
`arg 3 return`	return z
`arg 2 arg 1 con 1 arg 3 sub call 0`	$tak(z-1,x,y)$
`arg 1 arg 3 con 1 arg 2 sub call 0`	$tak(y-1,z,x)$
`arg 3 arg 2 con 1 arg 1 sub call 0`	$tak(x-1,y,z)$
`call 0 return`	return $tak(x,y,z)$

Assignment 6: Turing Machines(Given Mar. 19. 2012)

12pt

Problem 6.1 (Turing Machine)

Given the alphabet $\{0, 1\}$ and an arbitrarily initialized tape. Let the read/write head be on some position k. Define a transition table (with initial state $s1$) such that the machine runs for ever if (at start time) the entry on position $k+1$ is the same as on position k and terminates otherwise.

Assignment 8: Internet and protocols(Given Apr. 16. 2012)

12pt

Problem 8.1 (SMTP protocol)

1. What is SMTP? **Describe** the four basic commands that are sent to an SMTP server for sending an email (be careful at spelling them).
2. How do you tell the server that your email is finished?
3. Where is the subject of your email specified?

Solution:

1. (9p) SMTP is an email sending protocol (Simple Mail Transfer Protocol). The commands described on the slides are:
   (a) **helo** - we specify our identity
   (b) **mail from:** - tell our own address (careful with the ":")
   (c) **rcpt to:** - list the recipients of the email
   (d) **data** - specify the payload (message itself)
2. (2p) The email message is finished by transmitting a single dot (".") and nothing else on a line.
3. (1p) The subject is specified in the payload (specified after "data") by the email client.

Assignment 9: Web Search and Web Applications(Given Apr. 23. 2012)

6pt

Problem 9.1 (Web Crawlers)

Please explain what is a web crawler. Do not forget to mention the steps of its action cycle.

Solution: A web crawler is a computer program that browses the WWWeb in an automated, orderly fashion for the purpose of information gathering. Web crawlers are mostly used in web search engines.
The action cycle of a web crawler is the following:

- read web page
- report web page home
- find hyperlinks
- follow hyperlinks

Problem 9.2 (Web Applications) 6pt

Explain the terms web application, and web application framework. Give one example for each of them.

Solution:

1. A **web application**is a website that serves as a user interface for a server-based application using a web browser as the client
   examples: ebay.com, weather.com

2. A **web application framework** is a software framework for creating web applications
   examples: LAMP (linux, apache, MySQL, PHP/Python), XAMPP

Assignment 10: Internet Security and Introduction to Problem Solving(Given Apr. 30. 2012)

4pt

Problem 10.1 (Security by Encryption)
Please explain the terms

- one-way function
- trapdoor function

In addition, give one **example** of a candidate for a good one-way/trapdoor function, explaining how the function is computed, and how it is inverted.

Solution:

- A **one-way function** is a function that is "easy" to compute on every input, but "hard" to invert given the image of a random input.
- A **trapdoor function** is a one-way function is a one-way function that is easy to invert given a piece of information called the trapdoor.

For the example function, any of the examples on slide 380 or any other well-explained function works, e.g.:

- **Multiplication and Factoring:** The function f takes as inputs two prime numbers p and q in binary notation and returns their product. This function can be computed in $O(n^2)$ time where n is the total length (number of digits) of the inputs. Inverting this function requires finding the factors of a given integer N.
- **Modular squaring and square roots:** The function f takes two positive integers x and N, where N is the product of two primes p and q, and outputs x^2 div N. Inverting this function requires computing square roots modulo N ; that is, given y and N, find some x such that $x^2 \bmod N = y$. It can be shown that the latter problem is computationally equivalent to factoring N (in the sense of polynomial-time reduction).
- **Discrete exponential and logarithm:** The function f takes a prime number p and an integer x between 0 and $p1$; and returns the 2^x div p. This discrete exponential function can be easily computed in time $O(n^3)$ where n is the number of bits in p. Inverting this function requires computing the discrete logarithm modulo p; namely, given a prime p and an integer y between 0 and $p1$, find x such that $2^x = y$.

Problem 10.2 (Magic Square) 8pt

A magic square of order 3 is a 3×3 square grid, the cells of which are filled with disctinct integers, so that the sums of the 3 numbers in every row, every column, and in both diagonal, equal the same integer constant. Give a problem formulation as a tuple $P = \langle S, O, I, G \rangle$, describing the problem of solving the following magic square, which should contain only the digits from 1 to 9:

	5	

Note: If you find it hard or inappropriate to only give the mathematical formulation of some set, feel free to add a description in plain English.

Solution: Let the square be:

n_1	n_2	n_3
n_4	5	n_5
n_6	n_7	n_8

$P = \langle S, O, I, G \rangle$
$S = \langle n_1, n_2, n_3, n_4, 5, n_5, n_6, n_7, n_8 \rangle$
O : put permutation of the other 8 digits in the cells following some order, e.g. put digits in ascending order from left to right, row by row; check the sum; if it is not the same, go to the next state following the order
$I = \langle 5, \ , \ , \ , \ , \ , \ , \ , \ \rangle$
$G : n_1 + n_2 + n_3 = n_4 + 5 + n_5 = n_6 + n_7 + n_8 = n_1 + n_4 + n_6 = n_2 + 5 + n_7 = n_3 + n_5 + n_8 = n_1 + 5 + n_8 = n_3 + 5 + n_6 = 15$
or e.g. $G = \langle 4, 9, 2, 3, 5, 7, 8, 1, 6 \rangle$ for the square

4	9	2
3	5	7
8	1	6

Assignment 11: Tree searches(Given May. 07. 2012)

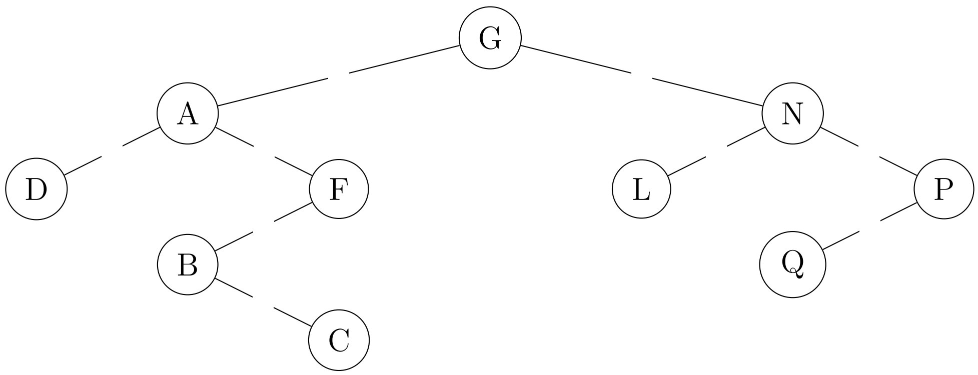
Problem 11.1 (Search algorithm) 6pt

Give the generic algorithm for search on trees (using pseudocode). You can use statements such as "pick the next unvisited sibling".

Solution: The algorithm is described on slides 411 and 417.

Problem 11.2 (BFS) 6pt

Apply BFS on the following tree (the goal is not node G!):

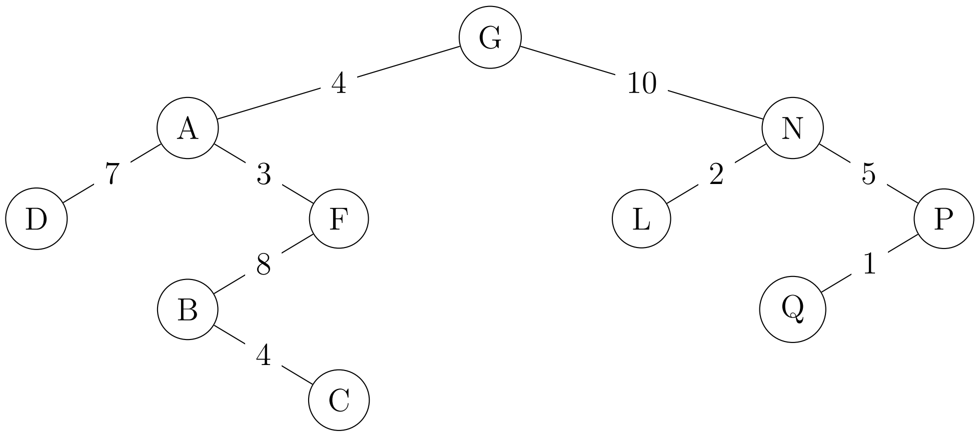


Solution: BFS: G, A, N, D, F, L, P, B, Q, C

Assignment 12: Tree searches(Given May. 14. 2012)

6pt

Problem 12.1 (UCS vs. IDS vs. Greedy)
Apply UCS, ID and Greedy search on the following tree:



The following heuristic values are given: $h(A) = 10$, $h(B) = 5$, $h(C) = 9$, $h(D) = 6$, $h(F) = 2$, $h(G) = 0$, $h(L) = 2$, $h(N) = 4$, $h(P) = 13$, $h(Q) = 3$

Solution:

1. UCS: G, A, F, N, D, L, B, P, Q, C (but E and P might be swapped)
2. IDS: G, G, A, N, G, A, D, F, N, L, P, G, A, D, F, B, N, L, P, Q, G, A, D, F, B, C, N, L, P, Q
3. Greedy: G, N, L, A, F, B, D, C, P, Q

Problem 12.2 (Astar vs. Greedy) 6pt

Shortly explain the principle of operation of the A* search. How does it differ from the Greedy search?

Solution: A* will expand the nodes in the fringe in an ascending order of the function f(node)=h(node)+g(node), where h(node) is the heuristic of the node and g(node) is the (current) distance from the initial node to this node. Greedy will expand only taking the heuristic into account.
