Algorithms and Data Structures (BADS)
Foundations of Computing – Algorithms and Data Structures (SGDS)

Exam 21 August 2014

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.)

Answering multiple-choice questions. 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.

number of checked boxes	0	1	2	3	4
points if correct answer checked		1	0.5	0.21	0
points if correct answer not checked	0	-0.33	-0.5	-0.62	

In particular, the best thing is to only check the correct answer, and the worst thing is to check all answers but the correct one. 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. Some questions are worth more points, or allow more than four choices; their points are scaled accordingly. For more details, read [Gudmund Skovbjerg Frandsen, Michael I. Schwartzbach: A singular choice for multiple choice. SIGCSE Bulletin 38(4): 34–38 (2006)].

(Just to make sure: a question that is not multiple-choice cannot give you negative points.)

Where to write. Please try to answer the exam by writing directly on the exam set. If you run out of space, or change your mind, then of course you can answer questions of a separate piece of paper. Just make it very clear (cross out everything and write "see separate paper, page 1" or something like that.) Question 4a needs to be answered on a separate paper anyway.

Typographic remark. I follow the typographic convention used impliclty in the course book that a one-letter Java variable, such as N, is typeset in italics like a mathematical variable in body text: N.

1. Analysis of algorithms

(a) (*1 pt.*) Which pair of functions satisfy $f(N) \sim g(N)$?

A $(N+1)(N+N+N+N)$ and $2N^2$ B $2N + 17$ and N

C $\log(3N)$ and $\log N$ D $(N \log N) + 16N^2$ and $(2N \log N) + 15N$

(b) (*1 pt.*) How many stars are printed when the following code is executed?

```
for (int i = N; i > 1; i--) StdOut.print("*");
```

A $\sim \log N$ B $\sim N$ C $\sim N \log N$ D $\sim \frac{1}{2}N^2$

(c) (*1 pt.*) How many stars are printed when I call $f(N)$ in the following code?

```
static void f(int K)
{   for (int i = 0; i < K; i++) g(K); }

static void g(int K)
{   for (int i = 0; i < K; i = 2*i) StdOut.print("*"); }
```

A logarithmic in N B linear in N C linearithmic in N D quadratic in N

(d) (*1 pt.*) What is the asymptotic running time of the following piece of code?

```
if (N < 1000)
  for (int i = 0; i < N; i++)
      for (int j = 0; j < N; j = j*2) A[i] = j;
else for (int i = 0; i < N; i++) A[i] = i;
```

A logarithmic in N B linear in N C linearithmic in N D quadratic in N

(e) (*1 pt.*) The next three questions consider a method f(int K) in a simple class F like this:

```
public class MyClass
{
  int N;
  public MyClass(int N) { this.N = N;}
  public void f(int K)
  {
    if (K == N) for (int i = 0; i < K; i++) StdOut.println("*");
    return;
  }
}
```

What is the running time of the following code in terms of N?

```
MyClass F = new MyClass(N);
F.f(N);
```

A constant   B logarithmic in N C linear in N D quadratic in N

(f) (*1 pt.*) What is the running time of the following code in terms of N?

```
MyClass F = new MyClass(N);
F.f(N*N);
```

A constant   B logarithmic in N C linear in N D quadratic in N

(g) (*2 pt.*) What is the running time of the following code in terms of N?

```
MyClass F = new MyClass(N);
for (int i = 0; i < N; i++) F.f(i);
```

A constant   B logarithmic in N C linear in N D quadratic in N

```
1  public class Z
2  {
3    String[]  names;
4    Integer[] numbers;
5    int N = 0;
6
7    public Z(int cap)
8    {
9      names   = new  String[cap];
10     numbers = new Integer[cap];
11   }
12
13
14   public Integer find(String S)
15   {
16     for (int i = 0; i < N; i++)
17       if (names[i].equals(S)) return numbers[i];
18     return null;
19   }
20
21   public void insert(String S, Integer M)
22   {
23     for (int i = 0; i < N; i++)
24       if (names[i].equals(S)) { numbers[i] = M; return; }
25     names[N]   = S;
26     numbers[N] = M;
27     N++;
28   }
29 }
```

Figure 1: Class Z.

2. Class Z. The next few questions all concern the class defined in fig. 1.

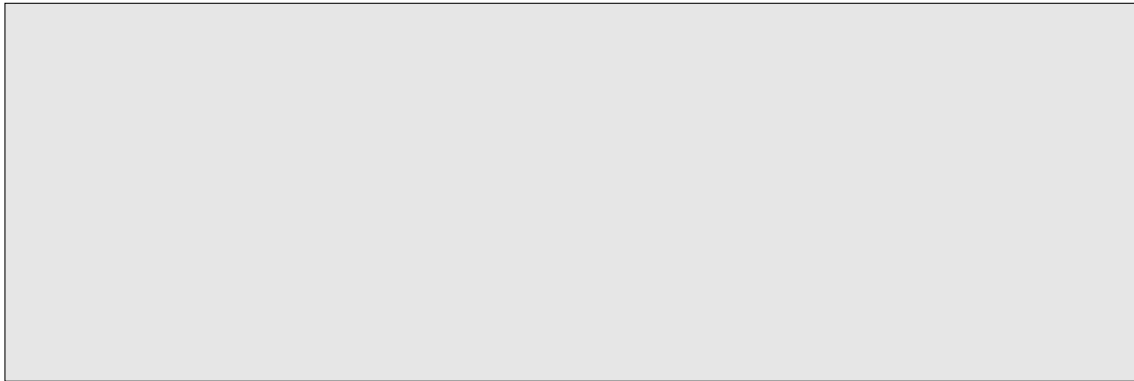(a) (*1 pt.*) Class Z behaves like which well-known data structure?

    A Stack.   B Symbol table.

    C Queue.   D Priority queue.

(b) (*1 pt.*) Draw the contents of the data structure after the following operations:

```
Z z = new Z(3);
z.insert("Bob",1);
z.insert("Charlie",1);
z.insert("Matthew",2);
z.insert("Bob",10);
```

(c) (*1 pt.*) What is the *total* running time of the following code (N is an int variable)

```
Z z = new Z(N);
for (int i = 0; i < N; i++) z.insert("Bob", i);
```

    A $O(\log N)$.   B $O(N)$.   C $O(N \log N)$.   D $O(N^2)$.

(d) (*1 pt.*) What is the *total* running time of the following code (N is an int variable)

```
Z z = new Z(N);
for (int i = 0; i < N; i++) z.insert(new String(i), i);
```

    A $O(\log N)$.   B $O(N)$.   C $O(N \log N)$.   D $O(N^2)$.

(e) (*1 pt.*) What is the running time of a single call to x.insert()?

    A Worst-case constant.

    B Amortised constant but worst-case linearithmic.

    C Worst-case linear.

    D Average-case logarithmic but worst-case quadratic.

(f) (*1 pt.*) Which claim is *true*?

    A The data structure uses constant space.

    B The keys are in sorted order.

    C insert may result in ArrayIndexOutOfBoundsException.

    D The data structure uses generic types.

(g) (*2 pt.*) Extend the code so that class Z implements Iterable<Boolean> by implementing

```
public Iterator<Boolean> iterator()
```
Write complete Java code; your implementation must be correct, efficient, and simple. Following the conventions of the book, implement `remove()` by the empty method.

(h) (*2 pt.*) Assume I have implemented a delete method using the convention of the book, like this:
```
public delete(String S) { insert(S,null); }
```

Implement the method `size()` that returns the number of elements in the data structure. Write complete java code. (You are graded on correctness, efficiency, and clarity.) What is the running time?

3. Operation of common algorithms and data structures.

(a) (*1 pt.*) Consider the following sequence of operations on a data structure, where a number i means insert(i) and "$*$" means `remove()` and then output the removed item. The data structure is initially empty.

$$1 \quad 12 \quad 5 \quad * \quad 3 \quad 7 \quad * \quad * \quad * \quad 2 \quad 4 \quad 13 \quad * \quad 14 \quad 15 \quad * \quad * \quad *$$

The output resulting from these operations is:

$$1 \quad 12 \quad 5 \quad 3 \quad 7 \quad 2 \quad 4 \quad 13$$
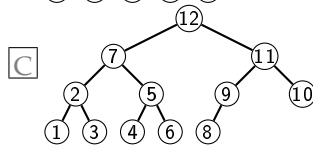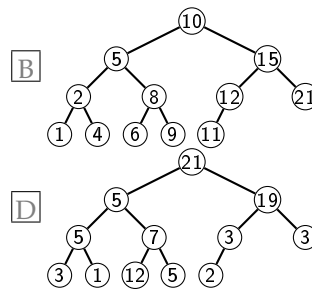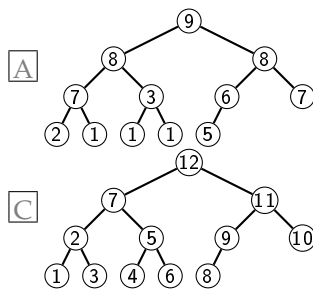
What is the data structure?
- A A symbol table
- C A queue
- B A stack
- D A priority queue

(b) (*1 pt.*) Which of the following trees is a binary search tree?
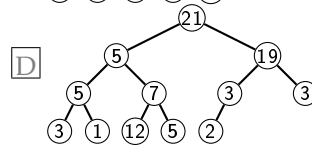
(c) (*1 pt.*) Insert the keys 7 6 5 4 3 1 2 in that order into a 2–3 tree. Draw the resulting tree.

(d) (*1 pt.*) I have sorted "aegon daeron dareon aenys aerys aelyx aemon daemon alyssa" using MSD string sort ([SW, section 5.1]). In the figure below, mark all characters that were examined by the MSD string sort with a circle.

```
aegon
aelyx
aemon
aenys
aerys
alyssa
daemon
daeron
dareon
```

(e) (*1 pt.*) Run (i) insertion sort and (ii) selection sort on the 4-letter input

```
E X A M
```

using the book implementations on p. 249 and p. 251 and stop after exactly 3 calls to the exchange method exch.
Write the resulting sequences. (Your answer consist of two sequences of exactly 4 characters, the first is for insertion sort, the second for selection sort.)

Insertion sort:

Selection sort:

(f) (*1 pt.*) Consider the key–value pairs

key	E	X	A	M	Q	U	E	S	T	I	O	N
value	0	1	2	3	4	5	6	7	8	9	10	11
hash	4	8	0	12	1	5	4	3	4	8	14	13

We use the hash function `key.hashCode() % 15` where A has `hashCode` 0, B has `hashCode` 1, and so on. To spare you the calculations, the hash values are given in the table above as well. The elements are inserted from left to right into an initally empty hash table of size 15 using linear probing. Draw the result in the style of the book [SW, p. 469]:

(g) (*1 pt.*) In which sense is a balanced binary search tree worse than a binary search tree with no balancing?

A It is more difficult to implement.

B It is harder to use from a client because of the invariants.

C It requires logarithmically more memory in the worst case.

D It cannot implement `rank()` and `select()` because of rebalancing.

(h) (*1 pt.*) This exercise is about tries. Consider the key-value pairs (`aegon`, 1), (`aenys`, 2), (`aerys`, 3), (`aelyx`, 4), (`aemon`, 5), (`daemon`, 6), (`alyssa`, 7). Insert them into an initally empty trie and draw the result:

4. Design of algorithms

The University of Maximegalon IV has switched to online teaching. Each of its *L* courses is available online, and you can take it any time you want. Each course comes with with a list of *prerequisite* courses that you must complete before the course. (For instance, you cannot take "Algorithms and data structures" unless you completed "Introduction to programming" and "Discrete mathematics" first.) You want learn Algorithmic Exobiology, and you want to do as little extra work possible.

The problem input. The problem input consists of *L* lines. Each line contains three tab-separated fields for the course code, a full course name, and a list of prerequisites (possibly empty).

Prog101	Introduction to programming	
AD	Algorithms and data structures	Prog101, DM
DM	Discrete mathematics	Calc, Alg
Calc	Calculus	
Alg	Algebra	
WA	Weird aliens: how to spot and avoid them	Run
Run	Running and personal fitness	
HR	Human resources and personell planning	
Econ101	Economics 101	Calc, Alg
AlgEx	Algorithmic Exobiology	AD, WA
MBA	Master of Business Administration	Econ101, HR
Lat	Latin	
Kl	Klingon	
Min1	Basic Minecraft	
Min2	Advanced Minecraft	Min1

Output. An ordered list of courses you need to take in that order. Example:

Calc, Alg, Prog101, DM, AD, Run, WA, AlgEx

(a) (*4 pt.*) Design an algorithm for the above problem.

You are encouraged to make use of existing algorithms, models, or data structures from the book, but please be precise in your references (for example, use page numbers or full class names of constructions in the book). Be short and precise. This question can be perfectly answered on half a page of text. (Even less, in fact.) If you find yourself writing much more than one page, you're using the wrong level of detail. However, it is a very good idea to include a drawing of a concrete (small) example. If you can avoid it, please do not write code. (However, some people have an easier time expressing themselves clearly by writing code. In that case, go ahead.)

You are evaluated on correctness and efficiency of your solution, and clarity of explanation.

(b) (*1 pt.*) Estimate the running time of your solution. (Make sure you tell me what the variables mean!)

Use page 9 for your answer to these questions.

(Answer to 4a and 4b go here.)