Algorithms and Data Structures (BADS/SGDS)

Exam 15 August 2017

Thore Husfeldt and Riko Jacob, 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. (Just to make sure: a question that is not multiple-choice cannot give you negative points.)

Where to write. Mark you answers to questions 1–3 on pages 8 and 9. If you really have to, you may use separate sheets of paper for these questions instead, but please be clear about it (cross out everything and write "see separate paper, page 1" or something like that.) Question 4 is answered on separate sheet(s) of paper anyway. For the love of all that is Good and Holy, write legibly. Hand in pages 8 and 9, and any separate sheet(s) of paper. Do not hand in pages 1–7, they will not be read.

Exam questions

1. Analysis of algorithms

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

A $f(N) = 2N$ and $g(N) = N + N + N$

[✓ B] $f(N) = 2N + 10 + N^2$ and $g(N) = N^2 + 10$

C $f(N) = 2N^2 + N$ and $g(N) = 2N^2 + N^2$

D $f(N) = 2N^2 + N$ and $g(N) = N^2 + N^3$

(b) (*1 pt.*) Which pair of functions satisfy $f(N) = O(g(N))$?

[✓ A] $f(N) = 17$ and $g(N) = 1$

B $f(N) = (N+1) \cdot (N+1)$ and $g(N) = N \log N$

C $f(N) = (\log N) \cdot (\log N)$ and $g(N) = 2 \log N$

D $f(N) = N^5 + 5N$ and $g(N) = 5N^4$

(c) (*1 pt.*) How many stars are printed?

```
for (int i = 1 ; i < N; i = i+2) StdOut.print("**");
```

Base revision 30b2f1a. . ., Mon Jan 22 16:07:16 2018 +0100, Riko Jacob @X260.

A $\sim 2\log_2 N$ [✓ B] $\sim N$ C $\sim N \log N$ D $\sim \frac{1}{2}N^2$

(d) (*1 pt.*) How many stars are printed? (Choose the smallest correct estimate.)

```
for (int i = 0; i < N/2; i = i+1)
     for (int j = 1; j < N/2; j = 2*j)
          StdOut.print("**");
```

A $O(\log N)$ B $O(N)$ [✓ C] $O(N \log N)$ D $O(N^2)$

(e) (*1C pt.*) What is the asymptotic running time of the following piece of code? (Choose the smallest correct estimate.)

```
if (N < 1000) for (int i = 0; i < N*N; i = i+1) A[i] = A[i/2];
else          for (int i = 0; i < N; i = i+1)   A[i] = A[i-1000];
```

[✓ A] linear in N B linearithmic in N C quadratic in N D cubic N

(f) (*1 pt.*) Find a recurrence relation for the number of arithmitic operations (additions and subtractions) performed by the following recursive method. (Choose the smallest correct estimate. The base case is $T(0) = 0$ in all cases.)

```
static int r(int N)
{
    if (N > 0) return r(N-1) + N + 1;
    else       return 1;
}
```

A $T(N) = T(N-1) + N + 1$ B $T(N) = T(N) + T(N-1)$
[✓ C] $T(N) = T(N-1) + 3$ D $T(N) = T(N-1) - N - 1$

(g) (*1 pt.*) Consider class `Amor` in Figure 1. Let N be an integer and define an object A by `Amor A = new Amor(N)`. Assume that `A.f()` has been called an arbitrary number of times. What is the worst-case running time of a single call `A.f()` ? (Choose the smallest correct estimate.)

[✓ A] Linear in N.
B Constant.
C Quadratic in N.
D Impossible to say from the information given.

(h) (*1 pt.*) Let A be defined as in the previous question. What is the runnig time of

```
for (int i = 0; i<N; i=i+1) A.f();
```

(Choose the smallest correct estimate.)

[✓ A] Linear in N.
B Constant.
C Quadratic in N.
D Impossible to say from the information given.

```
 1  public class Amor
 2  {
 3     int k, N;
 4
 5     public Amor(int N) { this.k=N/2; this.N = N; }
 6
 7     void f()
 8     {
 9       --k;
10       if (k < 0)
11          while (k < N) ++k;
12     }
13  }
```

Figure 1: Class `Amor`. In line 9, the symbols in front of *k* are two minus symbols.

```
    public static boolean f(int[] A, int[] B, int c)
    {
       for (int i= 0; i < A.length; i++)
           for (int j= 0; j < B.length; j++)
               if (A[i] + B[j] == c) return true;
       return false;
    }
```

Figure 2: Method f.

2. Method f. The next few questions all concern the method f in Fig. 2.

(a) (*1 pt.*) Assuming

```
int[] A = { 1, 10, 5 };
int[] B = { 3, 2 };
```

what is the value of `f(A,B,10)`?

(b) (*1 pt.*) Assuming

```
int[] A = { 1, 10, 5 };
int[] B = { 3, 2 };
```

give the smallest value for c such that `f(A,B,c)` is `true`.

(c) (*1 pt.*) What is the worst-case running time of f? (Let M denote `A.length` and let N denote `B.length`. Choose the smallest correct estimate.)

- A Constant.
- B Logarithmic in NM.
- C Proportional to $N + M$.
- [✓ D] Proportional to NM.

(d) (*1 pt.*) What is the best-case running time of f? (Let M denote `A.length` and let N denote `B.length`. Choose the smallest correct estimate.)

- [✓ A] Constant.
- B Logarithmic in NM.
- C Proportional to $N + M$.
- D Proportional to NM.

(e) (*1 pt.*) Which claim about f is false?

- A It uses linear space.
- B It raises an exception if either A or B are empty.
- C It does not change the contents of A or B.
- [✓ D] It is recursive.

(f) (*1 pt.*) Assume A and B are nonempty and consider the following expression:

```
f(A,B,...)
```

Write an expression in place of `...` that makes the above expression evaluate to `true`.

(g) (*1 pt.*) Consider the function

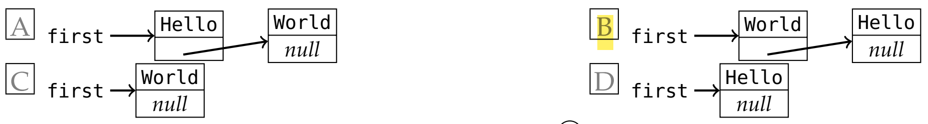
```
boolean g(int[] A) { return f(A,A,0); }
```

Explain what g computes by finishing the following sentence: "`g(A)` is true if and only if ..."

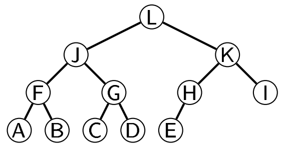
(h) (*3 pt.*) Describe how to rewrite f to make it faster. State the running time of your solution. (Your implementation may use more space. You may write code if you want, but don't have to.)

3. Operation of common algorithms and data structures.

(a) (*1 pt.*) Consider a Stack *S*, implementated as a linked list. Beginning with an empty data structure, assume I called `S.push("World"); S.push("World"); S.pop();`, and `S.push("Hello");`. How does the data structure look after these operations?



(b) (*1 pt.*) Consider the following tree:



Which statement is true?

[✓ A] The tree is in heap order, but not in search tree order.

B The tree is not in heap order, but in search tree order.

C The tree is both heap order and search tree order.

D The tree is in neither heap order nor search tree order.

(c) (*1 pt.*) Insert the keys `5 4 3 2 1` in that order into a binary search tree (without any rebalancing, using algorithm 3.3 in [SW]). Draw the result.

(d) (*1 pt.*) Assume I sort the letters `S E Q U E N C E` using selection sort. Which one of the following situations *can* arise at some time during the algorithm?

[✓ A] `C E Q U E N S E` B `C S E Q U E N E` C `S E Q U E C N E` D `E S Q U E N C E`

(e) (*1 pt.*) Consider a quick-find implementation of Union–Find with 5 elements. After some operations, the internal data structure looks like this:

`id[]`

0	1	2	3	4
0	1	2	2	4

Assume we now union elements 2 and 4. How does the resulting data structure look?

(f) (*1 pt.*) In which sense is it Weighted-Quick-union better than Quick-find?

A The union operation is asymptotically faster.

[✓ B] The amortized complexity of find is asymptotically faster.

C It can compute a minimum forest.

D It uses less space.

(g) (*1 pt.*) Insert the letters `P U T I N` in that order into a red–black BST as defined in the textbook (algorithm 3.4). Draw the resulting structure in the style of the book, use a fat edge to represent red links. (Your answer will be photocopied in black and white, so don't use fancy colours.)

(h) (*1 pt.*) The textbook implementation of binary heap is based on an array, rather than a linked structure of nodes. Why isn't such an array-based structure used for ordered search tree as well?

[✓ A] The implementation would use more than linear space because a search tree is not necessarily complete.

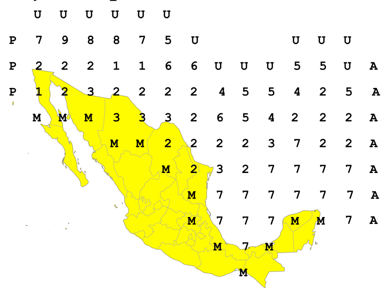
B Search trees also need to maintain the size of each node's subtrees, which cannot be stored in an array.

C Search trees can be much larger than heaps, so external memory considerations become more important.

D Ordered search trees must support the floor operation, which takes too much time in an array.

4. Design of algorithms You want to build a wall towards Mexico in the Southern United States. You have a map of the area, which is partitioned into squares. Each square can hold a piece of wall; the task is to connect the eastern and western shores with wall. Two squares are adjacent if they share a side to the north, south, east, or west. (In other words, the wall does not connect diagonally.) Due to various reasons (rivers, water, swamp, mountains, need to relocate a town, etc.), the sqares are more or less expensive to build on.

In the image below, you can see squares overlaid on a map of Mexico. The integers give the cost. The letters mean `A` for Atlantic, `P` for Pacific, `U` for the inner United States, and `M for Mexico`. No wall can be built on either of those four types of square. The task is to connect any of the `P` to any of the `A` using adjacent pieces of wall for as little total cost as possible.



Input

A space separated sequence of squares, line by line, in the obvious fashion. A full stop `.` is used for the empty squares. Note that the examples below are small, in general you cannot assume that the costs are single digits.

To fix notation, let r denote the number of rows in the input (i.e., the number of lines), and l the number of columns (i.e., the length of the longest line).

Output

The cost of the cheapest wall you can build.

Don't take 'wall' too literally. For instance, the area south of Florida goes across water, so this is probably a job for the US coast guard instead, rather than a bricklayer. Think of the wall pieces as 'making a route impossible to use.'

Also, don't let your knowledge of correct geography get in the of solving the exercise. An optimal solution to the example above loses San Diego and a good part of Florida, which would be undesirable in reality, but is not important for this exercise.

Example

Input:
```
. U U U U U U
P 7 9 8 8 7 5 U . . . U U U
P 2 2 2 1 1 6 6 U U U 5 5 U A
P 1 2 3 2 2 2 2 4 5 5 4 2 5 A
. M M M 3 3 3 2 6 5 4 2 2 2 A
. . . . M M 2 2 2 2 3 7 2 2 A
. . . . . . M 2 3 2 7 7 7 7 A
. . . . . . . M 7 7 7 7 7 7 A
. . . . . . . M 7 7 7 M M 7 A
. . . . . . . . M 7 M
. . . . . . . . . M
```
Output:
```
35
```

Input:
```
. U U U
P 1 1 3 A
P 3 1 1 A
. M M M
```
Output:
```
4
```

(a) (*3 pt.*) Describe an algorithm that solves the problem.

(b) (*2 pt.*) Because of shady deals with Russian entrepreneurs you found a cheaper way of constructing wall sections: Now each square costs the same. (Equivalently, you can pretend that all costs in the input are 1.) Describe an algorithm that solves this problem faster than your answer to 4a.

For both questions, explain your algorithm briefly and clearly, and state the running time of your algorithm in terms of the given parameters. You are strongly 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). Each 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.
Important: If you decide to view this problem as a graph problem, you *must* describe how the graph is constructed, and you *must* draw the *entire* graph corresponding to the smaller of the two examples.
You don't need to 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 solutions, and clarity of explanation.

Answers

This is where you mark your answers for questions 1–3. It is strongly preferred that you fit your answers on these sheets, except for 2h, 4a, and 4b. If you really must, you can use a separate sheet of paper instead. Please indicate that clearly.

Your name:

	1a	1b	1c	1d	1e	1f	1g	1h	2c	2d	2e	3a	3b	3d	3f	3h
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

2a

2b

2f `f(A, B,                     )` evaluates to *true*.

2g `g(A)` is true if and only if

2h on a separate piece of paper.

3c

3e

3g

4a, 4b *on a separate piece of paper.*