

Exam in EDAF05 Algorithms, Data Structures, and Complexity

May 29, 2018, 14-19

You may answer in English, på svenska, auf Deutsch, или по-русски.

Examiner: Jonas Skeppstedt

30 out of 60p are needed to pass the exam and your grade is $\lfloor \text{your score}/10 \rfloor$.

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, or old exams. 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 a pocket calculator, but I can't see how that would be of any use to you.

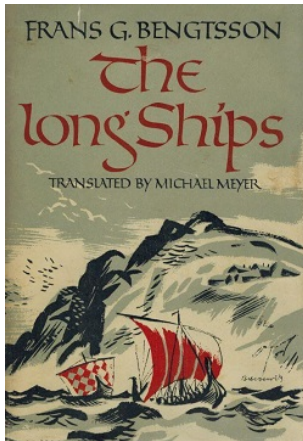
Filling out the exam. Some questions are multiple choice. Mark the box or boxes with a cross or a check-mark. If you change your mind, completely black out the box and write your answer's letter(s) in the left margin. In case it's unclear what you mean, I will choose the least favourable interpretation. In those questions where you have to write or draw something, I will be extremely unco-operative in interpreting your handwriting. So write clearly. If there is a way to misunderstand what you mean, I will use it.

Scoring. For the free form choice questions, you get between 0 and the maximum number of points for that question. Short, correct answers are preferred. You can, however, get a negative score on a multiple choice question. Each multiple choice question has exactly one correct answer. To get the maximum score for that question, you must check that answer, and only that. However, to reflect partial knowledge, you may check several boxes, in case you're not quite sure (this lowers your score, of course — the more boxes you check, the fewer points you score). If you check no boxes or all boxes, your score for that question is 0. If the correct answer is not among the boxes you checked, your score is negative, so it's better to not answer a question where you're on very thin ice. The worst thing you can do is to check all boxes except the correct one, which gives you a large negative score. For example, assume a question worth maximum 2 points has $k = 4$ possible answers (one of them correct).

- If you select only the correct answer, you receive 2 points.
- If you select 2 answers, one of which is correct, you receive 1 point.
- If you select 3 answers, one of which is correct, you receive 0.41 points.
- If you select no answer or all answers, you receive 0 point.
- If you select only one answer, and it is wrong, you receive -0.67 points.
- If you select 2 answers that are both wrong, you receive -1 point.
- If you select 3 answers that are all wrong, you receive -1.25 points.

As a special case, for a yes/no question, you receive 1, 0, or -1 points, depending on whether your answer is correct, empty, or wrong. The precise formula is: if the question has k choices, and you checked a boxes, your score is $\log(k/a)$, provided you checked the correct answer, and $-a \log(k/a)/(k-a)$ if you only checked wrong answers. Moreover, I have weighted the questions by relevance (not necessarily difficulty), and indicated the maximum points with each question. If you really care why this scoring system makes sense, read *Gudmund Skovbjerg Frandsen, Michael I. Schwartzbach: A singular choice for multiple choice. SIGCSE Bulletin 38(4): 34–38 (2006)*. For example, random guessing will give you exactly 0 points, at least in expectation.

Attack



After a successful expedition, Red Orm, Toke, and their friends, had, as usual, the problem of deciding how to split the treasures they had convinced their former owners to hand over to them. They put all valuables in boxes — hundreds of boxes — and wrote on each box its estimated value. As part of the deal with the English king who generously had collected the valuables for the Scanian vikings, they had agreed to take all the treasures, including the magic ones which actually had negative value. After all, Red Orm, Toke, and their friends were not greedy. They accepted these magic items without even considering making any use of their swords. Toke suggested they put the magic valuables in special boxes so everyone could see which boxes to, if useful, avoid, and clearly put the minus sign in front of the value.

After several hours, they had arranged all boxes in a long line and on top of each box, the value was written. Most had positive values but some had negative values.

Toke also suggested how to split the treasures. There were n boxes, and each box was numbered from 1 to n . Each viking, starting with Red Orm, was allowed to select two numbers i and j , and then take all boxes from number i to number j . Of course, $i \leq j$. After Red Orm had selected his boxes, these boxes were removed, the remaining boxes again put in a nice row, and renumbered, for Toke to make his selection, and so on.

Now, how should Red Orm make his selection? How should he select i and j and what is the value of his selection?

Example 1

Input:

box:	1	2	3	4	5
value:	-1	3	-2	4	-1

Output:

Boxes 2..4 give maximum value of 5

Example 2

Input:

box:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
value:	-3	4	-3	5	3	-3	4	0	-1	-9	2	4	3	2	1	-2

Output:

Boxes 11..15 give maximum value of 12

Planning

"So, what should we do now?" Red Orm asks Toke. "Well, after our visit to London, we have many ideas of where to go next. As a matter of fact, each ship crew in our fleet has discussed where we should go in the near future. Here is a summary of what each ship chief has proposed.", Toke said. "Let me see".

The suggested plans from the ship chiefs are described as follows: Each ship has a list of pairs of coastal cities to visit. A pair, such as A, B means it is necessary to visit A before visiting B . This can for instance be due to it is important to find resources in A for use when visiting B or it may be more difficult to visit A if B might warn A about what can happen. A list from a ship chief may look like this:

A, B
B, C
D, E
D, F

This means that A must be visited before B , which must be visited before C , and that D must be visited before any of E and F but it does not matter which of E or F is visited first, and it does not matter which of A or D is visited first either. Of course, no ship chief would propose a list like this:

A, B
B, A

Doing so would be very embarrassing.

"I see", Red Orm said. "But what should we do first? All ships must sail together. Toke: I don't want to see all these lists. Show me an itenary instead, if you can!", Red Orm told Toke. "Yes, I will, but do I have to take into account the distances to sail?", Toke asked. "No, no, just tell me if this is at all possible by showing me one itenary."

Toke then said, "So if we look at all the plans from every ship, we have a total of r requirements on the order in which two cities should be visited, and in total c cities, right?". "Of course" Red Orm replied.

Example 1

Ship 1 plan:

A, B B, C

Ship 2 plan:

A, C C, B

Output:

impossible

See next page for another example!

Example 2

Ship 1 plan:

```
A, B
B, C
```

Ship 2 plan:

```
A, C
D, E
```

Each of the following lines are examples of valid outputs:

```
A, B, C, D, E
D, E, A, B, C
A, D, B, E, C
```

It is sufficient to print exactly one valid output line.

Labs



It is soon summer break at the university, but at the Department of Ice Cream Tasting, the planning for the summer school is in full progress. The department will offer N courses and each course i needs $a(i)$ assistants. S students have applied for being lab assistants, and also indicated which courses they would like to work in. Due to an annoying department rule, a student is allowed to be an assistant in at most k courses. The number of students who will be assigned to work in course i is $w(i)$, and obviously $w(i) \leq a(i)$.

Help the department select which student should work in which course so that $\sum_{1 \leq i \leq N} w(i)$ is maximized.

Input format. The first line contains the number of courses, N , the number of students, S , and k . Then follow N lines with course number and the number of needed lab assistants for that course, i.e. $a(i)$. For simplicity we use a number instead of a course codes such as ICT101 which would just disturb. Then follow S lines with student name and a list of courses she or he has applied to.

Problem: employ as many students as possible, given the above constraint.

Example input:

```
4 6 2
1      2
2      3
3      3
4      2
Anna   1,2
Björn  1,2,3
Christina 1,3
Dag    2,3,4
Erik   3,4
Freja  1
```

Example output:

```
Anna      1,2
Björn     2,3
Christina 1,3
Dag       2,4
Erik      3,4
Freja
```

Distribution

As newly appointed head of the spin-off from the university, the Ice Cream Company, your first task is to set up ice cream stores in a certain number of Swedish cities. Due to company policy, newly established stores should not be located too close to each other. To be more precise, the minimum distance between stores must be 250 km. So you start checking city distances, city sizes, your budget, and start planning. Quick calculations indicate the budget makes it possible to open S stores this year, so that is your goal. You have made the decision to let any city with a population of at least 50,000 be a candidate for getting an ice cream store (for simplicity we ignore the fact that larger cities might be preferable).

Input format. The input consist of one line with the number of cities to consider, N , and the desired number of shops to open, S . Then follows a matrix of city distances. A city is identified below by a letter.

Problem. The problem is to determine, given the distances, if it is possible to open S stores? The output is either impossible or a list of cities.

Example 1

Input:

4 3				
	A	B	C	D
A	0	200	400	300
B	200	0	120	180
C	400	120	0	320
D	300	180	320	0

Output:

A, C, D

Example 2

Input:

4 3				
	A	B	C	D
A	0	200	100	300
B	200	0	120	180
C	100	120	0	220
D	300	180	220	0

Output:

impossible

Menus

A part of your ice cream store efforts is to create really nice looking menus. Considering that your ice creams are so novel and contain many healthy and tasty ingredients, the descriptions become quite elaborate — in fact, paragraphs of text. Excited about the prospect of eating a lot of ice cream, you unfortunately forgot about using L^AT_EX and instead decided to create a program to help you with the menus. Your motivation is that not only the ice creams but also the menus should look great. Therefore your main concern is the layout of space characters should be great. Nobody likes too uneven right margins. The rules for your menus are as follows:

- each ice cream description consists of a paragraph of n words,
- each word, $w(i)$ has $s(i)$ characters,
- you use exactly one space between each word,
- no word is to be split and put on separate lines,
- no line may contain more than 60 characters ("hello, world" counts as 12 characters),
- each line ends with zero or more spaces,
- except for the last line, you want to minimize the number spaces at the end of each line,
- it is fine to have any number of spaces on the last line, and
- you can use as many lines as you wish for a paragraph.

So your program must make decisions about which words should be on line one, line two, etc.

Let $x(i)$ be the number of trailing spaces on line i . Now you realize that your program needs to make an artistic decision of what looks good. Of course, you try out different alternatives and find that if you do as follows, the menus will look nice:

- minimize $\sum x(i)^3$ (except the last line).

To test your program, you try it on the following text you found on a famous company's web site.

Example

Input:

Incomparably creamy thanks to pure Swiss cream, an ice cream of authentic taste and timeless quality, in traditional Swiss style. Come and meet the well-loved Moevenpick ice creams. Embark on a journey of discovery!

Output:

Incomparably creamy thanks to pure Swiss cream, an ice cream
of authentic taste and timeless quality, in traditional
Swiss style. Come and meet the well-loved Moevenpick ice
creams. Embark on a journey of discovery!

Exam Questions

Pseudo code. Pseudo code is usually preferable to source code expressed in a programming language. The purpose, obviously, is to describe key ideas as simply as possible. Do not write any code to parse input but instead say for instance, "put each word from the input in an array" (if you want to do that).

Analysis of Algorithms

1. Let $f(n) = 2^{n+1}$. True or false?

(a) (1p) $f(n) = \Theta(2^n)$

A true B false

(b) (1p) $f(n) = \Omega(n^2)$

A true B false

(c) (1p) $f(n) = O(n^2)$

A true B false

2. (2p) Let $f(m, n) = (m + n)^2$. True or false? $f(m, n) = O(m^2 + n^2)$

A true B false

3. (2p) Let $f(a, n) = a^n$, and assume a is real and n is a positive integer. Ignore the limited precision of floating point arithmetic.

```
double f(double a, int n)
{
    double b;

    if (n == 0)
        return 1;
    b = f(a, n/2);
    if (n % 2 == 0)
        return b * b;
    else
        return a * b * b;
}
```

(a) The time complexity of $f(a, n)$ is — select the smallest correct estimate

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

4. **Graph search.** One of the problems can be solved using a simple graph search.

(a) (2p) Which one?

A Attack B Planning C Labs D Distribution E Menus

(b) (5p) Describe your solution. Use pseudocode.

(c) (3p) What is the running time of your solution?

5. **Divide-and-conquer.** One of the problems can be solved using divide-and-conquer.

(a) (2p) Which one?

A Attack B Planning C Labs D Distribution E Menus

(b) (6p) Describe your solution. Use pseudocode.

(c) (3p) What is the running time of your solution?

6. **Dynamic programming.** One of the problems is suitable to solve using dynamic programming.

(a) (2p) Which one?

A Attack B Planning C Labs D Distribution E Menus

(b) (7p) Describe your solution. Use pseudocode.

(c) (3p) What is the running time of your solution?

7. **Network flow.** One of the problems can be solved by a reduction to network flow.

(a) (2p) Which one?

A Attack B Planning C Labs D Distribution E Menus

(b) (4p) Explain how and illustrate with an example.

(c) (2p) What is the smallest correct running time? (see (d) below)

A $O(Cm)$ B $O(Cmn)$ C $O(m \log n)$ D $O((m+n)^2)$ E $O(4mn^2)$

(d) (2p) Which algorithm gives this time complexity, and what do C , m and n mean (if anything) in the problem?

8. **Computational complexity.** One of the problems is NP-complete¹.

(a) (2p) Which one? Call it P_1 .

A Attack B Planning C Labs D Distribution E Menus

(b) (2p) The easiest way to see this is to reduce from P_2 where P_2 is

A k -coloring B 3-SAT C Hamiltonian Cycle D Vertex Cover E Independent Set

(c) (1p) and prove: A $P_1 \leq_p P_2$ B $P_2 \leq_p P_1$

(d) (5p) Describe the reduction on a separate piece of paper. Do this both in general and for a small but complete example. In particular, be ridiculously precise about what instance is given, and what instance is constructed by the reduction, the parameters of the instance you produce (for example number of vertices, edges, sets, colors) in terms of the parameters of the original instance, what the solution of the transformed instance means in terms of the original instance, etc. Start your answer with the words "Given an instance to problem ...,"

¹If $P = NP$ then all these problems are NP-complete. Thus, in order to avoid unproductive (but hypothetically correct) answers, this question assumes that $P \neq NP$.