Instructions

What to bring. You can bring one book on computer science and one dictionary and you are allowed to have made any notes in them but no other papers are allowed. 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.
Transportation

Less than a month after the EU-elections in May, it was discovered that the newly elected members of the parliament had had too little focus on environmental concerns compared to their concerns for other less important issues. This resulted in a re-election in July in which Greta Thunberg’s hugely popular party won more than 70% of the seats.

One of her first decisions was to take measures to reduce the CO₂ emissions due to dirty transportations. Her proposal is simply that each city should replace its dirty with clean transportations by forbidding all types of transportations t to or from its neighboring cities if t pollutes too much. Therefore she wants to make as many cities as possible support her proposal. She quickly finds out, however, that all cities prefer to pollute the air in order to keep things as normal.

Her strategy then is to visit cities and encourage the population to support her. Since she does not want to visit all cities, and has limited time, she starts with Lund and asks students for assistance in writing a program to select cities to visit (in any order). Greta will then visit one city each weekend (going back to Brussels or Strasbourg between visits) until all direct connections between any two neighboring cities in the EU are clean.

**Problem:** Is it possible to make Europe free from dirty transportations before the next election in n weeks provided she visits one city each week and after a visit to a city, that city will support her proposal. Greta has a map with all direct connections between cities.

**Example 1**

*Input:*

| Weeks until next election: | 3 |
| Cities: | 4 |
| 1 -- 2 | |
| 2 -- 3 | |
| 3 -- 4 | |
| 4 -- 1 | |

*Output:*

| yes |

Not part of the output for the problem, but if Greta visits any three cities, and we assume each visited city changes its mind and supports her proposal, all dirty connections will be blocked.

**Example 2**

*Input:*

| Weeks until next election: | 2 |
| Cities: | 6 |
| 1 -- 2 | |
| 2 -- 3 | |
| 3 -- 4 | |
| 4 -- 5 | |
| 5 -- 6 | |
| 6 -- 3 | |

*Output:*

| impossible |
Cities

To help cities improving the air; Greta launched the "help your neighbor" project. Each city should become expert in one of two areas, use its knowledge for itself, and then help the neighbors. Some cities will focus on cleaning the air using filters denoted yellow cities, and the others will focus on reducing pollution, denoted blue cities.

**Problem:** Given a map with direct connections between cities, is it possible to assign each city either the yellow or blue label without two neighbors getting the same color?

**Example**

For both example inputs to Transportation the answer is yes, but not for the input below.

**Input:**

<table>
<thead>
<tr>
<th>Cities:</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -- 2</td>
<td></td>
</tr>
<tr>
<td>2 -- 3</td>
<td></td>
</tr>
<tr>
<td>3 -- 4</td>
<td></td>
</tr>
<tr>
<td>4 -- 5</td>
<td></td>
</tr>
<tr>
<td>5 -- 6</td>
<td></td>
</tr>
<tr>
<td>6 -- 1</td>
<td></td>
</tr>
<tr>
<td>6 -- 2</td>
<td></td>
</tr>
<tr>
<td>6 -- 3</td>
<td></td>
</tr>
</tbody>
</table>

**Output:**

impossible
Buses

“Autonomous self-driving buses may be a good option to investigate”, Greta told her Minister of Smart Technologies. “Sweden already has at least two pilot projects for this”, the minister replied. “I wonder how many rows with seats such buses should have?” Greta asked. “With too few there is no point with them and obviously the street curves and the number of interested passengers will give an upper limit. Please minister, go and select a suitable city and calculate how big buses would be useful to have there”, Greta told the minister.

“OK...” the minister answered, and immediately started to plan. “I will ask my staff to calculate the benefits of different bus sizes on the environment. Of course, the buses must be nice so people will use them...”, the minister thought.

A few days later the minister returned and presented the calculations. “Given a table, where the ith line shows the expected environmental benefits if the bus has i rows of seats, and we have in total n rows of seats available for this project, we can write a program to find out the best sizes of buses to use.”

“Great! Have you finished the program?” Greta asked. “Soon, I think”.

Problem

Given a table with n entries, 1 ≤ n ≤ 1000, where entry i is the expected benefit to the environment of building a bus with i rows of seats, determine which buses should be built in order to maximize the benefit, given that you can build any number of buses, limited only by the total number, n, of rows of seats.

Example

Input:

Total number of rows of seats: 5

<table>
<thead>
<tr>
<th>number of rows of seats in a bus</th>
<th>benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Output:

build 1 bus with 2 rows and 1 bus with 3 rows for a benefit of 9.

For example, instead building one bus with 5 rows would only give a benefit of 4.
Biking

After a few months in Strasbourg and Brussels, Greta calculated the environmental impact of moving staff between the cities every month, and discussed with party colleagues how they could use the resources consumed by this in a better way, or if this actually is the best for the world (which they at first glance suspect it is not). To create public interest in the question, they decided to bike from Brussels to Strasbourg, which is a distance of approximately 540 km using roads suitable for biking. They decided to always go south in the sense that being in one place $u$, they can only go to another place $v$ which is located at a coordinate which is south of $u$ (but an individual part of a road between $u$ and $v$ is allowed to be in the north direction). Further, they decided to bike on different roads in order to be seen by as many people as possible, and they want to go as many people as possible. There is no limit on the number of bikers which can reach the same place (the limit is only they should use different roads to and from that place). She turns to her newly appointed algorithm advisor to figure out what biking plans to make.

**Problem:** Make a biking plan following the rules described above so that as many bikers as possible can go from Brussels to Strasbourg.

*Example input:* A list of places with identifier and $(x, y)$ coordinate (where higher values of $x$ means more to the east and higher values of $y$ means more to the north). The first line is Brussels (i.e. A) and the last is Strasbourg (i.e. F), followed by direct road connections between places. All roads are bidirectional.

<table>
<thead>
<tr>
<th>A: 4,5</th>
</tr>
</thead>
<tbody>
<tr>
<td>B: 3,4</td>
</tr>
<tr>
<td>C: 3,2</td>
</tr>
<tr>
<td>D: 5,3</td>
</tr>
<tr>
<td>E: 6,2</td>
</tr>
<tr>
<td>F: 7,1</td>
</tr>
<tr>
<td>A -- B</td>
</tr>
<tr>
<td>A -- E</td>
</tr>
<tr>
<td>B -- C</td>
</tr>
<tr>
<td>C -- D</td>
</tr>
<tr>
<td>D -- E</td>
</tr>
<tr>
<td>D -- F</td>
</tr>
<tr>
<td>E -- F</td>
</tr>
</tbody>
</table>

*Example output:*

| 1 biker         |

A biker going from A to B to C will not be able to proceed to D since that is north of C, so only a biker going from A to E to F will reach F.
Skyline

“To use solar power more efficiently, we can put panels not only on roofs but also on the sides of buildings, especially since we are far away from the equator.” Greta said. Therefore, to facilitate the placement of new buildings and their solar panels, it is useful to have a drawing of all existing buildings. The Minister of Smart Technologies could not figure out how to do this, and surprised by this, Greta asked the minister to first try to make a simpler two-dimensional drawing. “OK, I will do that”, the minister replied.

Figure 1: Here are more complicated objects than in this problem, but the skyline is where black and white meet.

Problem: Given a set of buildings, which from the side all look like rectangles, produce a sequence of line segments so that the sequence shows the skyline of all buildings. Each building is described by its lower left and its upper right coordinates, in two dimensions. Two buildings can be overlapping, i.e. partly or fully covering another building. The input consists of lines with three numbers: the left x-coordinate, the right x-coordinate, and the right y-coordinate. The left y-coordinate is always zero. So 1,2,3 means a rectangle described by the coordinates \(x = 1, y = 0\) and \(x = 2, y = 3\). The right x-coordinate is always greater than the left x-coordinate and the right y-coordinate is always greater than zero.

Example input:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Example output:

0,0 - 0,3 - 1,3 - 1,4 - 2,4 - 2,3 - 3,3 - 3,4 - 6,4 - 6,0
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 \( f(n) = 2^n \). True or false?
   
   (a) (2p) \( f(n) = O(2^{2n}) \)

   \[ \begin{array}{c}
   \text{A true} \\
   \text{B false}
   \end{array} \]

2. Let \( f(n) = 2^{n+1} \). True or false?
   
   (a) (2p) \( f(n) = O(2^n) \)

   \[ \begin{array}{c}
   \text{A true} \\
   \text{B false}
   \end{array} \]

3. Let \( f(n) = 2^{2n} \). True or false?
   
   (a) (2p) \( f(n) = O(2^n) \)

   \[ \begin{array}{c}
   \text{A true} \\
   \text{B false}
   \end{array} \]

4. An advantage of Hollow heaps over traditional array based heaps is that Hollow heaps support inserting a new element in constant worst case time. True or false?

   (a) (1p) \( \text{A true} \) \( \text{B false} \)

5. Consider the function \( pow \), which computes \( a^n \) for \( n \geq 0 \).

```java
int pow(int a, int n) {
    if (n == 0)
        return 1;
    else if (n % 2 == 0)
        return pow(a, n/2) * pow(a, n/2);
    else
        return a * pow(a, n/2) * pow(a, n/2);
}
```

(a) (3p) The time complexity of \( pow \) is — select the smallest correct estimate.

\[ \begin{array}{c}
   \text{A} O(\log n) \\
   \text{B} O(n) \\
   \text{C} O(n \log n) \\
   \text{D} O(n^2)
   \end{array} \]
6. **Graph search.** One of the problems is suitable to solve using a simple graph search.
   (a) (4p) Which one?
      A Transportation  B Cities  C Buses  D Biking  E Skyline
   (b) (3p) Motivate your answer.

7. **Divide-and-conquer.** One of the problems is suitable to solve using divide-and-conquer.
   (a) (4p) Which one?
      A Transportation  B Cities  C Buses  D Biking  E Skyline
   (b) (2p) Explain briefly, in one sentence and no code, a simple algorithm with $O(n^2)$ time complexity to solve the problem.
   (c) (4p) Explain briefly, in one sentence and no code, a faster algorithm to solve the problem. You should motivate why this algorithm is faster than the $O(n^2)$ algorithm and state its time complexity.

8. **Dynamic programming.** One of the problems is suitable to solve using dynamic programming.
   (a) (4p) Which one?
      A Transportation  B Cities  C Buses  D Biking  E Skyline
   (b) (7p) Describe your solution in pseudocode.
   (c) (4p) What is the time complexity of your solution?

9. **Network flow.** One of the problems is suitable to solve by a reduction to network flow.
   (a) (4p) Which one?
      A Transportation  B Cities  C Buses  D Biking  E Skyline
   (b) (4p) Explain how and illustrate with an example.

10. **Computational complexity.** One of the problems is NP-complete\(^1\).
    (a) (4p) Which one? Call it $P_1$.
      A Transportation  B Cities  C Buses  D Biking  E Skyline
    (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) (3p) 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 ...."

\[ \text{Enjoy the summer!} \]

\[^1\text{If P = NP then all these problems are NP-complete. Thus, in order to avoid unproductive (but hypothetically correct) answers, this question assumes that P \neq NP.} \]
Notes, inaccuracies and disclaimer

- I obviously have no idea whatsoever about the opinions of Greta Thunberg about anything in this exam.
- Also obviously, this exam has no political intensions.