Algorithms Lab HS2016

Prof. Dr. Angelika Steger
Prof. Dr. Emo Welzl
Prof. Dr. Peter Widmayer

Andreas Baertschi, Daniel Graf, Dr. Michael Hoffmann,
Frank Mousset, Nemanja Škorić, Antonis Thomas,
Dr. Przemysław Uznański, Felix Weissenberger,
Manuel Wettstein

Objective: Master of Algorithms

Design efficient algorithms to solve "real world" problems.
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- an efficient implementation.

Note: usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

Objective: Master of Algorithms

Design efficient algorithms to solve "real world" problems.
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- an efficient implementation.

Note: usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

Objective: Master of Algorithms

Design efficient algorithms to solve "real world" problems.
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- an efficient implementation.

Note: usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

Objective: Master of Algorithms

Design efficient algorithms to solve "real world" problems.
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- an efficient implementation.

Note: usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

Example Problem

ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger
Prof. Dr. Emo Welzl
Prof. Dr. Peter Widmayer

Algorithms Lab

Exercise – *Clues*

Holmes and Watson are out on the streets to keep an eye on various people and places, hoping to obtain clues regarding criminal activities. Each of them carries a radio set and they want to coordinate through it as soon as something interesting happens. As the area of interest is quite large, Holmes and Watson cannot maintain a direct connection throughout the investigation. Instead they setup a network of radio stations to route the communication. In the following we use the term *clients* to refer to radio sets and radio stations collectively. All clients have the same operation range r so that they can communicate with every client in distance at most r.

For the actual communication there are four different frequencies available. Any client can receive on all frequencies. But in order to avoid interferences, any two clients that are in range of each other must send on different frequencies. Each of the two radio sets has one exclusive sending frequency assigned to it. This leaves two frequencies for the stations to work with. To keep the protocol simple, Holmes wants to assign one fixed frequency to each station so that the station sends on this frequency only. Is it possible to achieve such an assignment without generating any interferences? If so, which collections of clues can be routed within this network between Holmes and Watson?

The radio sets are "intelligent" in the sense that they automatically select the client to connect to. If the other radio set is in range, they connect to the other radio set. Otherwise—if any station is in range—they select the station with the strongest signal (closest client) to connect to. You may assume that this station is unique in all test instances.

Input The first line of the input contains the number $t \leq 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that contains three integers `n m r`, separated by a space and such that $1 \leq n, m \leq 9 \cdot 10^4$ and $0 < r < 2^{24}$. Here n denotes the number of stations, m denotes the number of clues, and r denotes the operation range of the clients.
- The following n lines define the positions $s_0, \ldots, s_{n-1}$ of the stations. You may assume that these positions are pairwise distinct.
- The final m lines define the m clues. Each clue is defined by two positions a_i and b_i, for $i \in \{0, \ldots, m-1\}$, where a_i describes the position of Holmes and b_i describes the position of Watson at the moment when this clue is obtained.

Each position is described by two integer coordinates `x y`, separated by a space and such that $|x|, |y| < 2^{24}$.

Output For each test case output a line with one character "y" or "n" per clue, that is, a string $c_0 c_1 \cdots c_{m-1}$ of m characters. For each $i \in \{0, \ldots, m-1\}$, the character c_i is "y", if and only if clue i can be routed within this network, as defined in the next paragraph.

Denote the set of stations by $S = \{s_0, \ldots, s_{n-1}\}$. A *network without interferences* on S corresponds to a map $f : S \to \{0, 1\}$ such that $f(u) \neq f(v)$, for all u, v with $||u - v|| \leq r$. A clue can be *routed* from a_i to b_i, if there exists a network without interferences on S and there exist $k \in \mathbb{N}$ and a sequence $t_0, \ldots, t_k$, such that

(1) $t_0 = a_i$, $t_k = b_i$, and $t_j \in S$, for $j \in \{1, \ldots, k-1\}$;

(2) $||t_j - t_{j-1}|| \leq r$, for all $j \in \{1, \ldots, k\}$;

(3) If $t_1 \neq b_i$ (and, thus, $t_{k-1} \neq a_i$), then t_1 is the (unique) client from S that is closest to a_i and t_{k-1} is the (unique) client from S that is closest to b_i.

Points There are four groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 20 points, you may assume that $n \leq 5'000$, $m = 1$, and $a_0 = b_0$. (Effectively, this reduces to the question of whether or not there exists a network without interferences.)
2. For the second group of test sets, worth 30 points, you may assume that there exists a network without interferences on S and $m \leq 20$.
3. For the third group of test sets, worth 30 points, you may assume that there exists a network without interferences on S.
4. For the fourth group of test sets, worth 20 points, there are no additional assumptions.

Corresponding sample test sets are contained in `test`i`.in/out`, for $i \in \{1, 2, 3, 4\}$.

Sample Input

```
2
2 3 2
0 0
2 0
-2 0 3 0
-2 1 2 -1
0 1 -1 2
3 1 2
0 0
2 0
1 1
3 0 1 1
```

Sample Output

```
yny
n
```

Course format

- This is a lab $\rightarrow$ most time spent working individually
- Tutorial: Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - remarks concerning problems from the previous week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- Problem of the week: Monday 17 – 19
  Exam like conditions: one problem, two hours to solve
- Consulting hours: Wednesday 19 – (after the tutorial)

Course format

- This is a lab $\rightarrow$ most time spent working individually
- Tutorial: Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - remarks concerning problems from the previous week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- Problem of the week: Monday 17 – 19
  Exam like conditions: one problem, two hours to solve
- Consulting hours: Wednesday 19 – (after the tutorial)

Course format

- This is a lab $\rightarrow$ most time spent working individually
- Tutorial: Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - remarks concerning problems from the previous week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- Problem of the week: Monday 17 – 19
  Exam like conditions: one problem, two hours to solve
- Consulting hours: Wednesday 19 – (after the tutorial)

Course format

- This is a lab $\rightarrow$ most time spent working individually
- Tutorial: Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - remarks concerning problems from the previous week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- Problem of the week: Monday 17 – 19
  Exam like conditions: one problem, two hours to solve
- Consulting hours: Wednesday 19 – (after the tutorial)

Prerequisites

See the course website:

`http://www.cadmo.ethz.ch/education/lectures/HS16/algolab/prereqs`

Most important:

Strategies	Brute force, greedy, divide & conquer, dynamic programming, backtracking, binary search
Data structures	Array, stack, set, queue, tree, heap, hashtable
Graph algorithms	DFS, BFS, minimum spanning tree, Dijkstra
Graph concepts	Directed graph, coloring, matching, topological sorting, (strongly) connected components, matchings

Occasionally we do a recap, but we do *not* give full explanations.

Synopsis

Three parts:

Fundamentals BFS, DFS, greedy, binary search, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

Advanced Algorithms Dynamic programming, exponential algorithms, network flows, LP/QP, Delaunay triangulations

Exam preparation select & combine the above

Three libraries at your disposal:

STL C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

Synopsis

Three parts:

Fundamentals BFS, DFS, greedy, binary search, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

Advanced Algorithms Dynamic programming, exponential algorithms, network flows, LP/QP, Delaunay triangulations

Exam preparation select & combine the above

Three libraries at your disposal:

STL C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

Synopsis

Three parts:

Fundamentals BFS, DFS, greedy, binary search, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

Advanced Algorithms Dynamic programming, exponential algorithms, network flows, LP/QP, Delaunay triangulations

Exam preparation select & combine the above

Three libraries at your disposal:

STL C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

Synopsis

Three parts:

Fundamentals BFS, DFS, greedy, binary search, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

Advanced Algorithms Dynamic programming, exponential algorithms, network flows, LP/QP, Delaunay triangulations

Exam preparation select & combine the above

Three libraries at your disposal:

STL C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

Problems

- Every Wednesday: new set of problems (3 or 4)
- Students submit their solutions within one week
- Automated grading/feedback by an online judge
- We provide solutions for a few selected problems.

Testat

No testat. – However . . .

6 ECTS credits correspond to 180 working hours.

How to get help

If you cannot solve a problem, you have two options.

- Your best bet for quick help are our **forums**, where other students or an assistant can help you out.
- Alternatively, use the **consulting hours** after the tutorial.
- **Important:** try to solve the problems on your own. In the exam you will not have access to the forums.

For administrative or technical problems (e.g., with the judge or moodle) use

Contact

algolab@lists.inf.ethz.ch

All other questions should be discussed on the forums.

Central Course Web Site: Moodle

There you can:

- download slides of the tutorials,
- download problem sheets,
- submit solutions, and
- discuss problems with your colleagues in the forums.

Login

- `https://moodle-app2.let.ethz.ch/login`
- NETHZ account / NETHZ password
- Enrolment Key (for the judge) *exchangeofinformation*

Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam ($\rightarrow$ `judge/doc`).
- Some testsets are **hidden**, you do not see the results for these.

NEW: clarifications: (PotW and exam only) questions about the problem statement should be submitted as *clarification requests* on the judge.

Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam ($\rightarrow$ `judge/doc`).
- Some testsets are **hidden**, you do not see the results for these.

NEW: clarifications: (PotW and exam only) questions about the problem statement should be submitted as *clarification requests* on the judge.

Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam ($\rightarrow$ `judge/doc`).
- Some testsets are **hidden**, you do not see the results for these.

NEW: clarifications: (PotW and exam only) questions about the problem statement should be submitted as *clarification requests* on the judge.

Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam ($\rightarrow$ `judge/doc`).
- Some testsets are **hidden**, you do not see the results for these.

NEW: clarifications: (PotW and exam only) questions about the problem statement should be submitted as *clarification requests* on the judge.

Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam ($\rightarrow$ `judge/doc`).
- Some testsets are **hidden**, you do not see the results for these.

NEW: clarifications: (PotW and exam only) questions about the problem statement should be submitted as *clarification requests* on the judge.

Computer Rooms

Solutions to problems (incl. the problem of the week) can be handed in from anywhere.

We have reserved computer rooms on Mondays which you can use when solving the problem of the week.

- In CAB: CAB H 56, CAB H 57
- In HG: HG E 26.1
- Everywhere else on your own laptop (+Internet connection)

Exam / Grades

The grade

is based only on the exam.

- 2 sessions of 6 hours each
- HG computer rooms, no custom hardware
- Submission/judging of programs exactly as in semester
- Very similar to the PotW
- Documentation on `judge.inf.ethz.ch/doc`
- NEW: No additional material
- Repetition: exam only once per year!

What will be on `judge/doc`?

- C++ reference (`cppreference.com`)
- Boost manuals
- CGAL manuals
- Tutorial slides (possible exceptions will be marked explicitly)
- (simple) index for searching

As a general rule, solutions will not be available there.

Break

After the break:

- Sample problem
  - Show you the complete procedure of solving & submitting a problem to the judge
- Course website
  - Forum etiquette