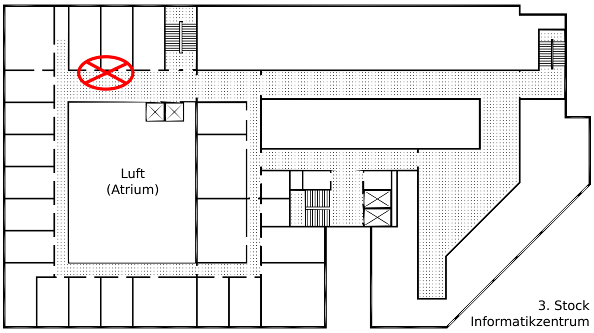
Abteilung Algorithmik **Summer term 2016**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Phillip Keldenich

Online Algorithms
4^{th} Homework Assignment, 8^{th} of June, 2016

Solutions are due on Wednesday, the 22^{nd} of June, 2016, until 11:30 in the homework cupboard. You can also hand in your solution in person before the small tutorial begins. External students can hand in the homework via e-mail to `c.rieck@tu-bs.de`.

Please make sure your homework submissions are clearly labeled using your name and/or matriculation number.



Exercise 1 (Looking around a corner): In the lecture, we considered the problem of looking around a corner with scan costs. In this situation, a robot it located at a known distance d from a corner. Distances are measured in travel time, normalized by scan time, i.e. travelling distance d takes the same time as d scans. Behind the corner, at an unknown angle, there is some hidden object. The goal is to move to the corner, performing scans along the way until the object is seen, minimizing the time taken for travelling and scanning.

We restrict ourselves to travelling on a polygonal chain inscribed in a semicircle of diameter d defined by the corner C and our starting location s. Giving any fixed competitive factor c we want to achieve, we can compute the distances x_i travelled between the ith and the $(i-1)$st scan using a recursive formula as presented in the lecture:

$$x_{i+1} = \underbrace{c(1+d_i)}_{\text{offline cost}} - \underbrace{(1+i)}_{\text{scan cost}} - \underbrace{\sum_{j=1}^{i} x_i}_{\text{travel cost}} \quad .$$

Here, d_i denotes the distance from the starting point to the ith scan point. For the first step, set $d_0 = 0$ and thus $x_1 = c - 1$. The distance d_i can be computed from the angles $\varphi_i = 2\arcsin(\frac{x_i}{d})$ covered by the ith segment as

$$d_i = d \sin\left(\frac{1}{2}\sum_{j=1}^{i} \varphi_j\right) .$$

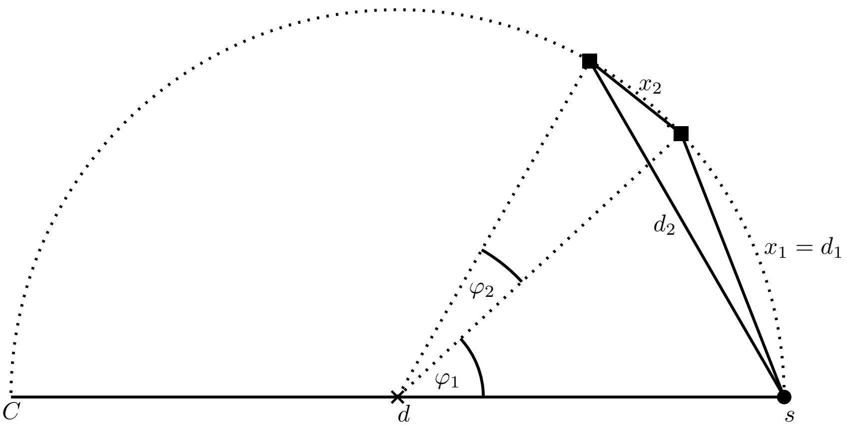


Figure 1: The first two steps of looking around a corner

Depending on the parameter c, this recursion either reaches the corner (after finitely many steps, the angle covered reaches or exceeds π) or collapses (distance we are allowed to travel becomes non-positive). In the first case, a competitive ratio of c is achievable by travelling on a semicircle; otherwise, it is not.

a) Why does the robot have to move to the corner?

b) Prove that moving to the corner directly is not c-competitive for any $c \geq 1$.

c) Implement a program, using a common programming language of your choice, that is able to check, for a given pair c, d, whether a factor of c is achievable for a starting distance of d.

d) Use your program in conjunction with the *bisection method* to find the best achievable competitive ratio for $d = 30$. Hint: For $d = 40$, the factor c satisfies $2.0015 < c < 2.0016$.

(5+5+15+15 points)

Exercise 2 (Star search): Consider the following generalization of line search: There is a crossing c with m rays emanating from it. We start at the crossing c and search for an item hidden on one of the rays. We know that the distance at which the item is hidden satisfies $D > 1$.

We can use a generalization of the cow search strategy: We number the rays by $1, \ldots, m$ in clockwise order, starting from an arbitrary ray, and visit them periodically in this order, so that the sequence of visits looks like $1, \ldots, m, 1, \ldots, m, \ldots$. As in the cow search strategy, we increase the distance we travel in one direction on the corresponding ray in the ith step after each unsuccessful try, using step length

$$x_i := \left(\frac{m}{m-1}\right)^i$$

for the ith step. Prove that this strategy cannot achieve a competitive factor better than

$$c := 1 + 2\frac{m^m}{(m-1)^{m-1}}.$$

(20 points)