RWTH Aachen **WS 2017**
Theoretical Computer Science Exercise 1
Rossmanith–Dreier–Kuinke 24.10.2017

Exercise for Analysis of Algorithms

Exercise T1

Consider the following algorithm that computes the maximum element in an array of positive integers. We assume that all elements are pairwise different and that each permutation occurs with equal probability.

```
int maxElem(int *a, int N)
{
  int i,max;

  max = -1;               /*  1  */
  for (i=0; i<N; i++)     /*  2  */
    if (a[i] > max)       /*  3  */
      max = a[i];         /*  4  */
  return max;             /*  5  */
}
```

- How often are the lines 3 and 4 executed in the *worst case* (in the *best case*)?
- What is the probability that this worst case occurs?
- How often are the lines 3 and 4 executed in the *average case*?

Exercise H1

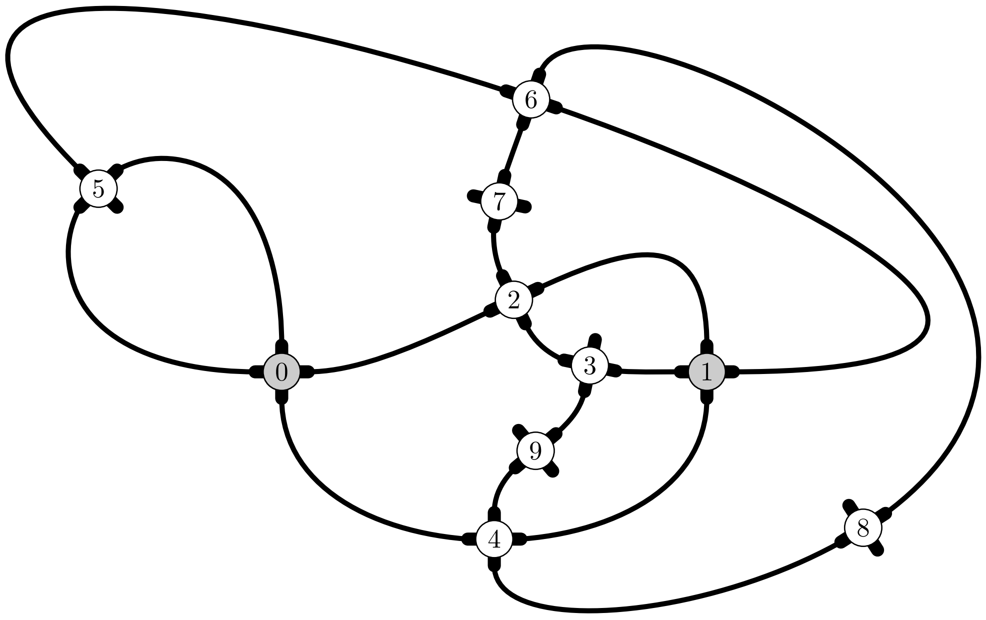
RWTH Aachen University has an exclusive contract with the well-known *Uranus Corporation* on the delivery of canal and pump supplies. Unfortunately, the responsible person failed to notice that Uranus sells only the "one-fits-all" product KKuRPSE (Kanalkopplungsundregulierungspumpstationeinheit). Such a KKuRPSE is a cylinder with four connectors placed north, east, south, and west of the cylinder (see the figure).

Now, the Institute for Tunnel and Canal Construction (ITCC) is conducting some research: The researchers first place n KKuRPSEs on a big green. Then they start to connect KKuRPSEs as follows: They digg a new canal between two arbitrary connectors. The new canal cannot cross another existing canal, of course. They then place a new KKuRPSE somewhere into this new canal, such that exactly two connectors on opposite sides of the new KKuRPSE are now attached to the canal.

The question is: How often can this connection procedure be repeated depending on the number n of initial KKuRPSEs on the green?

Example

Assume the researchers start with two KKuRPSEs, here labeled 0 and 1. Then a couple of connection steps are executed, where each new KKuRPSE receives consecutive numbers until no more connections are possible.



Exercise H2

Two natural numbers $m \neq n$ are called *friendly*, if the sum of all factors of m equals n — or the other way round. A son and a father wrote the following programs, that compute friendly numbers. What is their running time?

Son

```
#include <iostream>
int e[150000];
int realdiv(int a) {
  int n=0;
  for(int i=1; i+i<=a; i++)
    if(a%i==0) n+=i;
  e[a] = n;
  return n;
}
main() {
  for(int i=0; i<150000; i++) {
    int a = realdiv(i);
    if(a >= i) continue;
    if(e[a]==i) {
        std::cout << i << " "
        << realdiv(i) << "\n";
    }
  }
}
```

Father

```
#include <stdio.h>
#define N 1000000
int factorsum[N];
int main() {
  int i;
  for(i=1; i<N; i++) {
    int p=i;
    while(p<N) {
      factorsum[p] += i;
      p += i;
    }
  }
  for(i=1; i<N; i++) {
    int a = factorsum[i]-i;
    if(a<i && i==factorsum[a]-a)
      printf("%d %d\n", a, i);
  }
  return 0;
}
```