Analysis of Algorithms

Problem 3-1

In this exercise, we consider Prim's Algorithm, which computes a minimum spanning tree. The input to this algorithm is a graph $G = (V, E)$, a weight function on the edges $w : E \to \mathbf{R}$ and a starting node r.

```
 1  for each u ∈ V do
 2      key[u] ← ∞
 3      π[u] ← NIL
 4  key[r] ← 0
 5  M ← V
 6  while (M ≠ ∅) do
 7      u ← min-from(M)
 8      for each v ∈ neighbors(u) do
 9          if (v ∈ M) ∧ (w(u, v) < key[v]) then
10              π[v] ← u
11              key[v] ← w(u, v)
```

Construct the control flow graph, a spanning tree in the control flow graph, the fundamental cycles, a corresponding linear system of equations and a solution to this system.

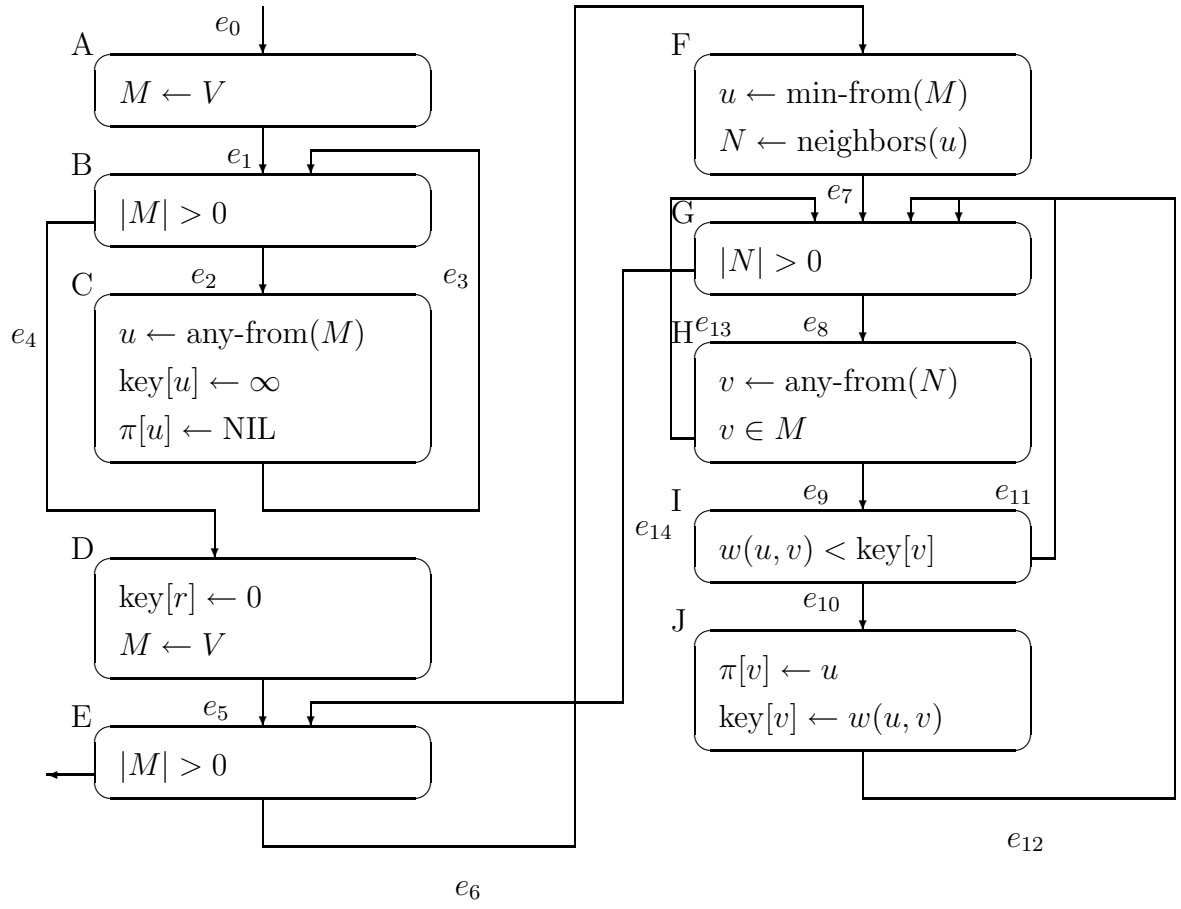
Solution:

The flow diagram is depicted below. The **for**-loops were changed, since the initializing and iteration condition must be separated.

We choose the spanning tree $e_1, e_2, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}$. This yields the following fundamental cycles:

$$\begin{aligned} C_0 &= e_0 + e_1 + e_4 + e_5 \\ C_3 &= e_3 + e_2 \\ C_{11} &= e_{11} + e_8 + e_9 \\ C_{12} &= e_{12} + e_8 + e_9 + e_{10} \\ C_{13} &= e_{13} + e_8 \\ C_{14} &= e_{14} + e_6 + e_7 \end{aligned}$$

We now use standard linear algebra to find a good set of blocks whose number of visits we need to compute: By E_i, $0 \leq i \leq 14$, we denote the number of times the program flow visits the edge e_i. With each fundamental cycle above we identify a vector C_i. Then the



E_i can be written as a linar combination of the fundamental cycles, i.e.,

$$(C_0, C_3, C_{11}, C_{12}, C_{13}, C_{14}) \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{pmatrix} = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \\ E_6 \\ E_7 \\ E_8 \\ E_9 \\ E_{10} \\ E_{11} \\ E_{12} \\ E_{13} \\ E_{13} \end{pmatrix}$$

for appropriate values of $\lambda_1, \ldots, \lambda_6$. We select six independent rows and obtain the

equation

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{pmatrix} = \begin{pmatrix} E_0 \\ E_2 \\ E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \end{pmatrix} = \begin{pmatrix} 1 \\ C \\ I-J \\ J \\ H-I \\ G-H \end{pmatrix}.$$

This means, we only need to compute the values of C, G, H, I, J for a complete analysis ($E_0 = 1$ is trivally known), and then all other values can be derived: We see that $E_0 = E_1 = E_4 = E_5$, $E_2 = E_3$, $E_6 = E_7 = E_{14}$, $E_{10} = E_{12}$. This implies $A = 1$, $B = E_1 + E_3 = C + 1$, $D = E_4 = 1$, $E = E_5 + E_{14} = 1 + G - H$, $F = E_6 = G - H$.

Exercise 3-2 The following program prints a lot of numbers. How many?

```
void snapl(int n) {
 if ( n == 0 || n == 2 ) return;
  if ( n == 1 ) std::cout << 42 << std::endl;
  else {
    snapl( n - 2 );
    snapl( n - 3 );
    snapl( n - 2 );
    snapl( n - 3 );
    snapl( n - 2 );
  }
}
```

Solution:
We get the recurrence $a_n = 3a_{n-2} + 2a_{n-3}$ for $n > 3$ and $a_0 = a_2 = 0$, $a_1 = 1$.
The characteristic polynomial is $z^3 - 3z - 2$. We need to find its roots.
Obviously, -1 is a root of $z^3 - 3z - 2$. By polynomial division we obtain $z^2 - z - 2$ as the remaining polynomial. Since -1 is also a root of this polynomial, we need to apply polynomial division again, which yields $z - 2$. The remaining root is 2.
The closed representation for a_n is of the form $\lambda(-1)^n + n\mu(-1)^n + \nu 2^n$. Using a_0, a_1 and a_2, we obtain

$$\begin{aligned} a_0 = 0 &= \lambda + \nu && (1) \\ a_1 = 1 &= -\lambda - \mu + 2\nu && (2) \\ a_2 = 0 &= \lambda + 2\mu + 4\nu. && (3) \end{aligned}$$

The first equation implies $\lambda = -\nu$ and thus

$$\begin{aligned} a_1 = 1 &= -3\lambda - \mu && (4) \\ a_2 = 0 &= -3\lambda + 2\mu. && (5) \end{aligned}$$

A simple computation now shows $\mu = -\frac{1}{3}$, which implies $3\lambda = 2\mu$ and thus $\lambda = -\frac{2}{9}$. The closed form is therefore

$$-\frac{2}{9}(-1)^n - n\frac{1}{3}(-1)^n + \frac{2}{9}2^n = \frac{2+3n}{9}(-1)^{n+1} + \frac{2^{n+1}}{9}.$$

Homework Assignment 3-1 (10 Points)

Consider the following program: The input to this program is an array $a[0, \ldots, n-1]$

```
int sel_sort ( int a[], int n ) {
  for ( int i = 0; i < n; ++i ) {
    int min = i;
    for ( int j = i; j < n; ++j ) {
      if ( a[j] < a[min] ) {
        min = j;
      }
    }
    int temp = a[i];
    a[i] = a[min];
    a[min] = temp;
  }
}
```

that contains n pairwise distinct integer keys in random order.

a) Explain how this program sorts the given array.

b) Analyse how often each instruction of the program is executed on average depending on n.

c) There is only one instruction whose analysis is not trivial. Which one is it?

Make a table for small values of n by hand that lists the results for this instruction. Compare the table entries with the results from your closed formula that you obtained in b).

Solution:

The algorithm is, as indicated by the name, selection sort: it finds the minimal element of the yet-to-be-sorted part and appends it to the sorted part by exchange.

For part b), the outer loop is execute n times, therefore each statement not in the inner loop is executed that often. The if-statement is executed exactly $\frac{n(n+1)}{2}$ times, which leaves the min $= j$ statement.

The first comparison ($j = i$ to i) always fails, therefore the expected number of executions (cf. Problem 1-2) then is

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} \frac{1}{j-i+1} = \sum_{i=0}^{n-1} \sum_{j=1}^{n-i-1} \frac{1}{j+1} = \sum_{i=0}^{n-1} (H_{n-i} - 1) = n(H_{n-1} - n)$$

Homework Assignment 3-2 (10 Points)

Solve the following recurrence: Let $b_1 = b_2 = b_3 = 1$ and

$$b_n = 3b_{n-1} - 4b_{n-2} + 12b_{n-3} \text{ for } n > 3.$$

Solution:

Here the characteristic polynomial is $x^3 - 3x^2 + 4x - 12$. We guess the first root $x_0 = 3$. Using polynomial division, we gain $x^2 + 4$. The two other roots are thus $x_1 = 2i$ and $x_2 = -2i$. Hence, the solution is of form $b_n = \lambda 3^n + \mu_1(2i)^n + \mu_2(-2i)^n$. Using the known values for $n = 1, 2, 3$, we obtain a system of equations with three variables λ, μ_1, μ_2 that can be solved with the standard methods.