Exam Algorithms Theory

Winter 2008/09

Name : .....

Matriculation number : .....

Program of study (Studiengang) : .....

Result (do not write here!)

Task 1		of 15	
Task 2		of 15	
Task 3		of 15	
Task 4		of 15	
Task 5		of 15	
Task 6		of 15	
Task 7		of 15	
Bonus points		of 15	
Sum		of 90	Top 6

Grade:

The exam consists of 7 tasks and 20 pages. From these 7 tasks and the bonus points of the exercise class the 6 highest numbers will account for the result. The maximum reachable number of points is 90; the exam is passed with at least 45 points.

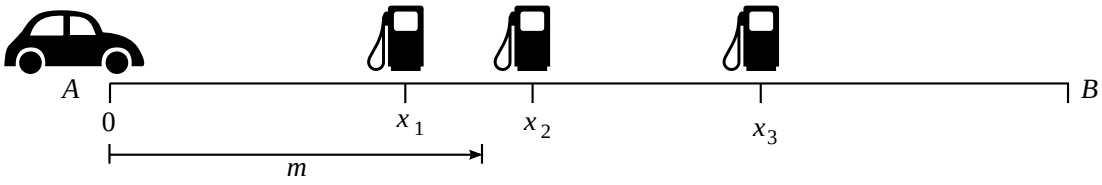
Please write your matriculation number on each sheet.

Please write your solution in the boxes below the tasks. If the space is not sufficient, you can ask for additional sheets.

Non-programmable calculators and an A4 sheet with hand-written notes (double-sided) are permitted.

Task 1: Greedy Algorithms **15 points**

You are driving on a highway from town A to B. Your gas tank holds fuel for m kilometers. A list of all gas stations with their positions $x_1, x_2, ..., x_n$ along your way is given. You want to minimize your stops and reach the destination if possible.



1. Design a greedy algorithm, that computes a minimal number of stops.

2. Show that your algorithm produces an optimal solution.

Matriculation number:

Task 2: FFT **15 points**

Compute the product of the two polynomials $p(x) = 7x + 3$ and $q(x) = -x + 2$ using the Fast Fourier Transformation.

1. Compute the FFT of $p(x)$ and $q(x)$ using an appropriate choice of k (for the k-th roots of unity).

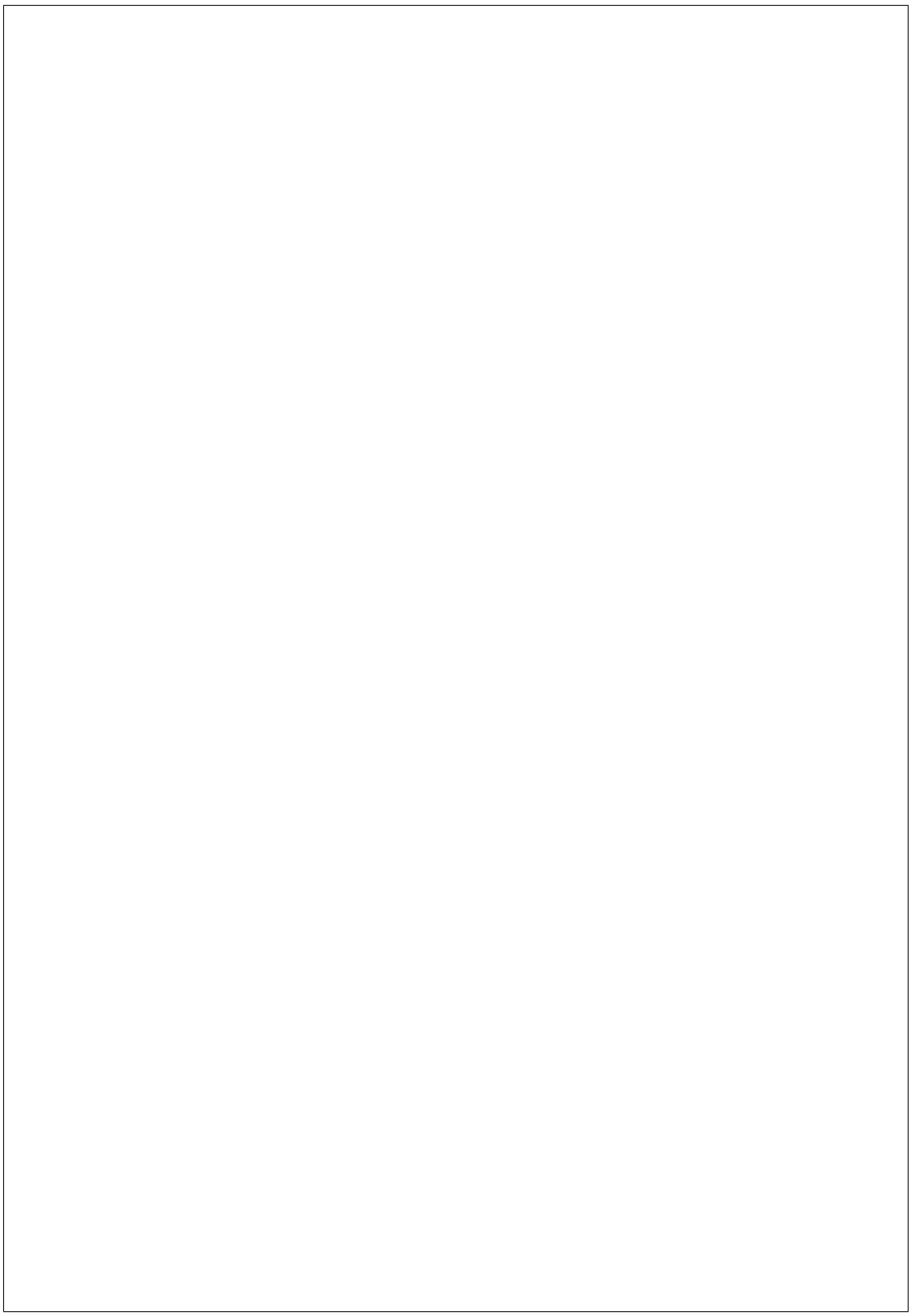
2. Give the point-value representation of pq at the k-th roots of unity.

3. Compute the interpolation by using the FFT algorithm.

4. Check the correctness of your result.

Specify all (recursive) calls of FFT algorithm as well as the outputs used during the execution.

Matriculation number:



Matriculation number:

Task 3: KMP and Boyer-Moore **15 points**

Given the pattern P = ABABBCABACABABB and the text
T = BARBABABBCABACABABBCABABBCABABBCABACABABBCDBBA

1. Compute the next-array for the pattern P.

A	B	A	B	B	C	A	B	A	C	A	B	A	B	B

2. Execute the KMP-Algorithm to search the pattern in the text. Present each shift of the pattern during the search for all occurrences by inserting the matching prefixes into the following table.

B	A	R	B	A	B	A	B	B	C	A	B	A	C	A	B	A	B	B ...

C	A	B	A	B	B	C	A	B	A	B	B	C	A	B	A	C	A	B ...

A	B	B	C	D	B	B	A											

Matriculation number:

3. Compute the last occurence table of the Boyer-Moore Algorithm (Version 1)

4. Execute the Boyer-Moore Algorithm Version 1 (without GSF) to search the pattern in the text. Present each shift of the pattern during the search for all occurrences by inserting the matching suffixes into the following table.

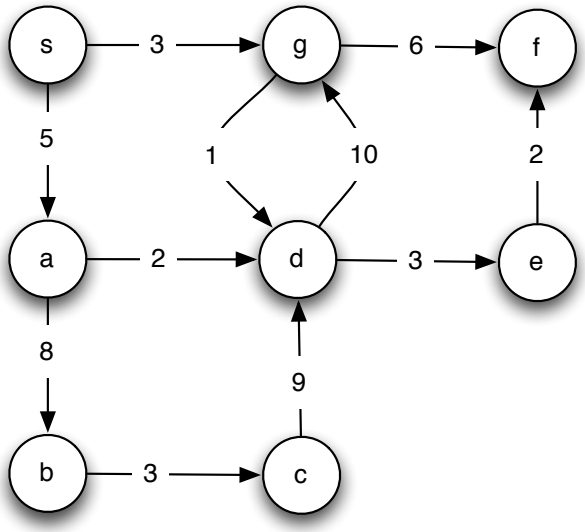
B	A	R	B	A	B	A	B	B	C	A	B	A	C	A	B	A	B	B ...

C	A	B	A	B	B	C	A	B	A	B	B	C	A	B	A	C	A	B ...

A	B	B	C	D	B	B	A											

Task 4: Shortest Paths **15 points**

1. Consider the following directed graph G. Describe an algorithm in pseudo code which finds single source shortest path trees of G. Apply this algorithm on G. Start from vertex **s** and describe each step of the algorithm.
2. Does your algorithm consider negative edge weights correctly? If not, describe a way to improve it accordingly.



Matriculation number:

Task 5: RSA **15 points**

Let $p = 19$, $q = 23$ and $e = 13$. Calculate d using *Extended Euclid* and determine the public key P and the private key S of RSA. Encrypt the message $M = 42$ using the *Fast Exponentiation* algorithm.

Matriculation number:

Task 6: Suffix Trees **15 points**

1. Which operations can be efficiently provided by *suffix trees*?

2. Create a suffix tree for the word T = RELIEFPFEILER and insert all suffix links.

3. How much memory is necessary to store a word of length n in a suffix tree.

Matriculation number:

Task 7: Edit distance **15 points**

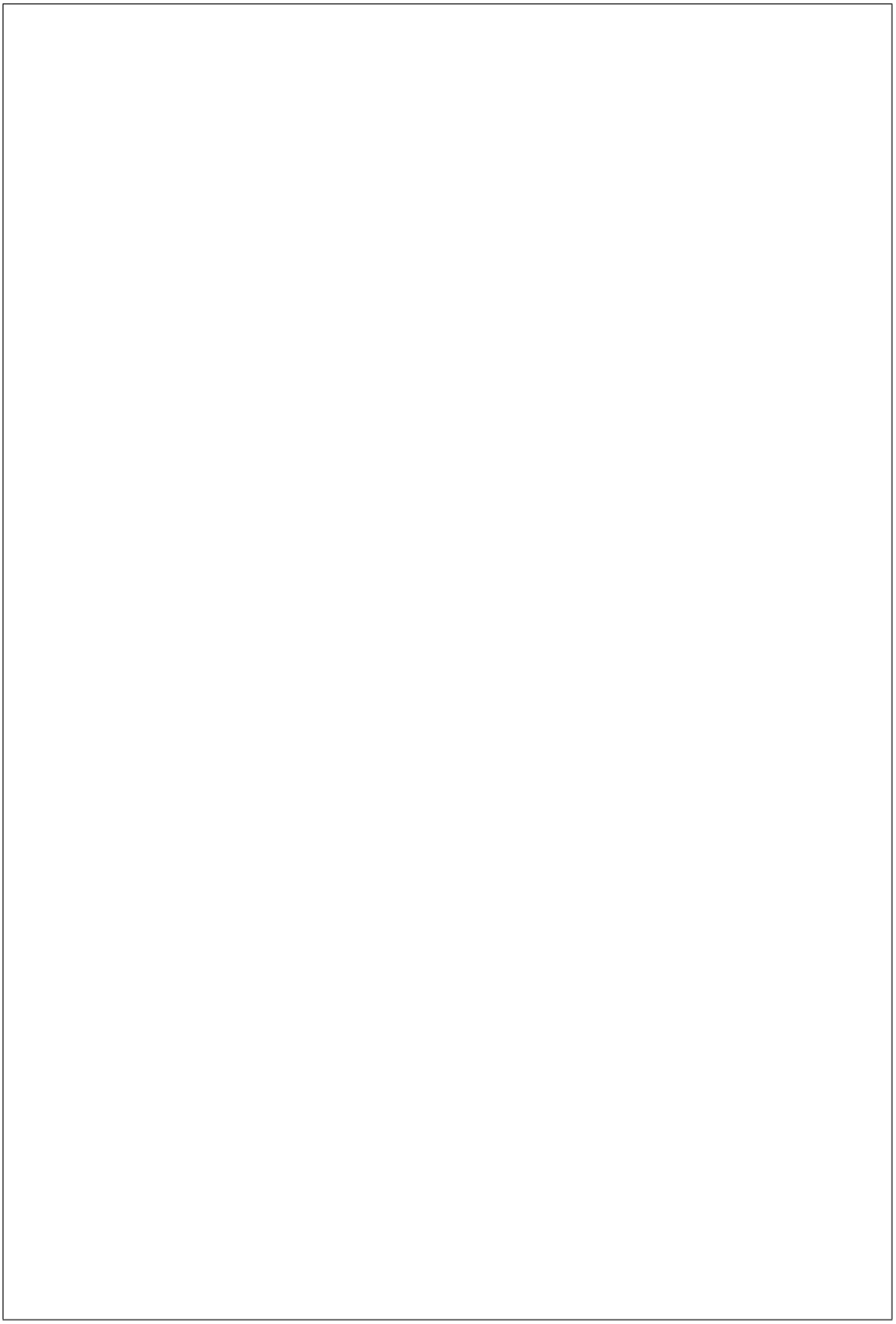
1. Describe an algorithm to determine the *edit distance* of two strings A and B.

2. Calculate the edit distance of the words $A =$ POLARSHADE and $B =$ PALISADE using this algorithm.

		P	O	L	A	R	S	H	A	D	E
P											
A											
L											
I											
S											
A											
D											
E											

		P	O	L	A	R	S	H	A	D	E
P											
A											
L											
I											
S											
A											
D											
E											

3. Describe an algorithm to determine the *approximate matching* of two strings A and B.



4. Find an approximate matching of the substring $S =$ SARA in the text string $T =$ PHILIPPUSPARACELSUS.

		P	H	I	L	I	P	P	U	S	P	A	R	A	C	E	L	S	U	S
S																				
A																				
R																				
A																				

		P	H	I	L	I	P	P	U	S	P	A	R	A	C	E	L	S	U	S
S																				
A																				
R																				
A