Exam Programming Fundamentals

February 26, 2025 (18:30-21:30)

- You can earn 90 points. You will get 10 points for free. So, you can obtain 100 points in total, and your exam grade is calculated by dividing your score by 10.
- This exam consists of 5 problems. The first problem consists of multiple choice questions. The problems 2, 3, and 4 are programming exercises in C. Problem 5 is about program correctness in Dafny. All problems are assessed by the Themis judging system. For each of the problems 2, 3, and 4, there are 10 test cases. Each of these test cases is worth 10% of the points.
- In the last hour of the exam, all inputs for the programming problems (2-4) will be made visible in Themis. Note that you can see the test cases of a problem only after having made a submission for that problem.
- Note that manual checking is performed after the exam. It is not allowed to use precomputed answers in your program. If this is detected by manual inspection, then all points for that exercise will be subtracted.
- This is an open book exam. You are allowed to use the ansi C book and the Dafny book. The pdf of the reader is available in Themis, as well as pdfs of the lecture slides. Any other documents are not allowed. You are allowed to use previous submissions that you made to Themis for the labs. Moreover, from Themis you can download an implementation of merge sort that you may use in your solutions.

Problem 1: Time complexity (30 points)
In this problem the specification constant `N` is a positive integer (i.e. `N>0`). Determine for each of the following program fragments the *sharpest upper limit* for the number of calculation steps that the fragment performs in terms of `N`. For a fragment that needs `N` steps, the correct answer is therefore $O(N)$ and not $O(N^2)$ as $O(N)$ is the sharpest upper limit.

1. ```
   int s = 0;
   for (int i = 0; i < N; i++) {
     for (int j = 0; j < N; j++) {
       s = s + i*j;
     }
   }
   ```
   (a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

2. ```
   int s1 = 0, s2 = 0, p = N;
   while (p > 0) {
     s1 += (p%10)*(1 - p%2);
     s2 += (p%10)*(p%2);
     p /= 10;
   }
   ```
   (a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

3. ```
   int s = 0, p = 0;
   for (int i=0; i*i < N; i++) {
     for (int j=0; j*j < N; j++) {
       s++;
     }
   }
   while (s > 0) {
     p += s;
     s--;
   }
   ```
   (a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$
   ```

4.
```
int i = 3, j = N;
while (i < j) {
  if (N%i == 0) {
    i *= 2;
  }
  j--;
  i++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

5.
```
int i = 1, s = 0;
while (i < N/i) {
  s += i;
  i++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

6.
```
int s = 0;
for (int i=N; i >= 0; i--) {
  for (int j=i; j > 0; j/=2) {
    s++;
  }
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

7.
```
int s = 0, i = 0, j = N;
while (i < j) {
  if (s%2 == 0) {
    i = 2*i + 1;
  } else {
    j = j/2 - 1;
  }
  s++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

8.
```
int s = 0, i = 0;
while (s < N) {
  s += i;
  i++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

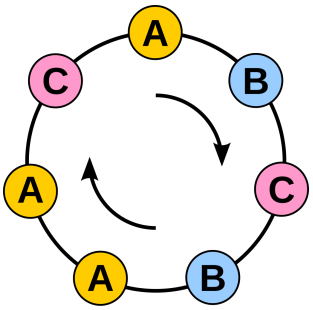
9.
```
int n = 0, d = 0;
while (d < N*N) {
  d = d + 2*n + 1;
  n++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

10.
```
int s = 0;
for (int i=0; i < 2*N; i+=2) {
  for (int j=i*i; j > 0; j/=2) {
    s++;
  }
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

11.
```
int s = 0, t = 0;
for (int i = 0; i < N; i+=2) {
  s += i;
}
while (s > 0) {
  t += s;
  s -= 2;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

12.
```
int s = 0, t = 0;
for (int i = 0; i < N; i+=2) {
  s += i;
}
while (s > 0) {
  t += s;
  s /= 2;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

Problem 2: Clockwise string rotation (15 points)
Consider the string `"ABCBAAC"`. We can rotate its characters one position in clockwise direction to obtain the string `"BCBAACA"`.



From the above figure, we see that the following set of strings is equivalent under clock wise rotation:

{`"ABCBAAC"`, `"BCBAACA"`, `CBAACAB"`, `"BAACABC"`, `"AACABCB"`, `"ACABCBA"`, `"CABCBAA"`}

The input consists of two strings. Each string consists of at most 1000 characters from the set {`'A'`, `'B'`, `'C'`}. The output of the program must be `YES` if the two strings are equivalent under clockwise rotation, and `NO` otherwise.

Example 1:
input:
ABCBAAC
ACABCBA
output:
YES

Example 2:
input:
ABCBAAC
CAABCBA
output:
NO

Example 3:
input:
ACCCAACCAA
CAAACCCAAC
output:
YES

Problem 3: Maximal duplicate sum (15 points)
The input of this problem is an array of positive `int`s. At least one element occurs more than once. The output must be the number that occurs multiple times, for which the sum of all its occurrences is maximal. In case this number is not unique, the largest number should be printed. For example, for the input array `[7,3,3,3,7,3,7,3,3,3,2]` the maximal sum is 21. The output must be `7`, because `21=7+7+7=3+3+3+3+3+3+3` and $7 > 3$.
The input of this problem consists of a single line containing a positive integer `n` followed by a colon (`:`). Next follows an array of `n` positive `int` values. You may assume that the maximal sum fits in a standard `int` (i.e. no overflow).

Example 1:
input:
```
11: 7 3 3 3 7 3 7 3 3 3 2
```
output:
```
7
```

Example 2:
input:
```
12: 3 7 3 3 3 7 3 7 3 3 3 2
```
output:
```
3
```

Example 3:
input:
```
7: 3 2 4 3 1 2 2
```
output:
```
3
```

Problem 4: Two player game (15 points)
Given an array with non-negative numbers, two players play the following game. Player 1 and player 2 take turns, with player 1 starting first. Both players start the game with a score of 0. At each turn, the player takes one of the numbers from either end of the array (i.e., the first or last number) which reduces the size of the array by one element. The player adds the chosen number to his/her score. The game ends when there are no more elements in the array. The winner is the player that has the highest score (or the game ends in a draw in case both players have the same score).

The input consists of two lines. The first line contains a positive integer n (where $1 \leq n \leq 25$), which is the length of the array. The next line contains the array itself. The output of your program must be `YES` if player 1 wins the game, and `NO` otherwise (i.e. player 2 wins, or the game ends in a draw). You may assume that both players are playing optimally.

Example 1:
input:
```
3
1 5 2
```
output:
```
NO
```

Example 2:
input:
```
4
1 5 2 3
```
output:
```
YES
```

Example 3:
input:
```
5
12 19 2 13 7
```
output:
```
NO
```

Problem 5: Dafny (15 points)
From Themis, you can download the file `problem5.dfy` which contains the following code fragment:

```
method mod3(x: nat) returns (m: nat)
  ensures m == x%3
{
  var n : nat;
  n := x;
  m := ??;
  while (??)
  invariant m<3 && ??
  {
    m := ??;
    n := n/10;
  }
}
```

The method `mod3` takes a natural number `x`, and returns the result `x%3` (i.e. the remainder of division of x by 3). The algorithm is based on the following observation. If `x` equals `10*n+m` then the remainder equals the remainder of `n+m`, because `10*n==9*n + n` and `9` is clearly a multiple of `3`.
At several locations in the code there are question marks. Replace the questions marks by expressions, such that Dafny accepts the program fragment. Note that you are not allowed to add or remove extra statements, or change pre/post-conditions. You are only allowed to replace the question marks by suitable expressions. [Note: if you use the visual studio code environment, please make sure you save your file (Ctrl-s) before submitting it to Themis!]

This page can be used as scratch paper

This page can be used as scratch paper

This page can be used as scratch paper

This page can be used as scratch paper