Resit Programming Fundamentals

February 28, 2024 (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 i = 0, s = 0;
   while (s < N*N) {
     i++;
     s += 2*i + 1;
   }
   ```
   (a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

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

3. ```
   int s = 0, i = 0;
   while (s < N) {
     for (j = 0; j < 10; j++) {
       s += i + j;
     }
     i += 10;
   }
   ```
   (a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$
   ```

4.
```
int s = 0, n = N;
while (n > 0) {
  n = (n%2 ? n/2 : n - 1);
  s += n;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

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

7.
```
int s = 0, t = 0;
for (int i = 0; i < N; i++) {
  s += i;
}
while (s > 2) {
  s = 1 + s/2;
  t++;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

8.
```
int i = 0, j = N;
while (j - i > 1) {
  int m = (i + j)/2;
  i = (m*m <= N ? m : i);
  j = (m*m > N ? m : j);
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

9.
```
int s = 0;
for (int i = 0; i < N; i++) {
  int k = 2 + i%5;
  for (int j = 1; j < N; j*=k) {
    s += i + j;
  }
}
```
(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 = 1; i < N; i++) {
  for (int j = 0; j < 10; j++) {
    for (int k = N-j; k >= 0; k--) {
      s += i + j + k;
    }
  }
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

11.
```
int i = 0, j = 0, s = 0;
while (i*j < 2*N) {
  i = i + 1;
  j = j + 2;
  s = s + i + j;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

12.
```
int i = 1, s = 0;
while (i*i < N) {
  i = 2*i;
}
while (i > 0) {
  for (int j = 0; j < i; j++) {
    s += i + j;
  }
  i--;
}
```
(a) $O(\log N)$ (b) $O(\sqrt{N})$ (c) $O(N)$ (d) $O(N \log N)$ (e) $O(N^2)$

Problem 2: Special Numbers (15 points)
In this exercise, we call a number n *special* if n, $n-1$, and $n+1$ can all be written as the product of two distinct primes.

For example, $n = 34$ is special because $n = 34 = 2 \times 17$, $n - 1 = 33 = 3 \times 11$, and $n + 1 = 35 = 5 \times 7$. On the other hand, 33 is not special because $32 = 2 \times 2 \times 2 \times 2 \times 2$.

The input for this problem consists of an integer m (where $1 \leq m \leq 10^7$). The output must be the number of special numbers n with $1 \leq n \leq m$.

Example 1:
input:
```
33
```
output:
```
0
```

Example 2:
input:
```
34
```
output:
```
1
```

Example 3:
input:
```
100
```
output:
```
3
```

Problem 3: Pair Difference (15 points)
The input for this problem consists of a positive integer k, and an integer array x without duplicates. The output must be the number of pairs (i, j) such that $x[i] < x[j]$ and $x[j] - x[i] \leq k$.

The input of this problem consists of two lines. The first line contains the integer k. The second line contains a positive ineger `n` followed by a colon (`:`). Next follows the array x of `n` non-negative `int` values.

Example 1:
input:
```
3
6: 1 2 3 4 5 6
```
output:
```
12
```

Example 2:
input:
```
6
10: 25 33 3 46 39 61 57 12 44 64
```
output:
```
5
```

Example 3:
input:
```
2
9: 3 1 10 13 9 18 19 12 16
```
output:
```
6
```

Problem 4: Minimal Palindromic Inserts (15 points)
A *palindrome* is a word that reads the same backward as forward. Some examples are `level`, `madam`, and `radar`.

The word `sublevels` is clearly not a palindrome, but we can make it a palindrome by inserting letters. For example, `subSlevelsBUS` and `sublevelsLEVELBUS`, and `sublevelBUs` are all palindromes that can be produced by inserting letters (which are denoted in uppercase for readability). The last example requires only two insertions, which is minimal for the word `sublevels`.

The input for this problem is a single word, consisting of at most 32 lower case letters. The output must be the minimal number of inserts needed to transform the input word into a palindrome.

Example 1:
input:
`sublevels`
output:
`2`

Example 2:
input:
`radars`
output:
`1`

Example 3:
input:
`madam`
output:
`0`

Problem 5: Dafny (15 points)
From Themis, you can download the file `problem5.dfy`. At several locations in the file, 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 program statements, or change pre/post-conditions. However, you are allowed to insert **assert** statements if needed.

```
method problem5(x: int, y: int) returns (z: int)
requires y >= 0
ensures z == x*y
{
  var a := x;
  var b := y;
  z := 0;
  while ??
    invariant ??
  {
    if ?? {
      z := ??;
      b := b - 1;
    }
    b := b/2;
    a := 2*a;
  }
}
```